

VŠB – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra informatiky

# **Tvorba prostředí pro ovládání provozně technických funkcí v Smart Home Care**

## **Development of Software Tool for Operational and Technical Functions Control in the Smart Home Care**

# Zadání diplomové práce

Student:

**Bc. Jana Belešová**

Studijní program:

N2647 Informační a komunikační technologie

Studijní obor:

2612T025 Informatika a výpočetní technika

Téma:

**Tvorba prostředí pro ovládání provozně technických funkcí v Smart Home Care**  
**Development of Software Tool for Operational and Technical Functions Control in the Smart Home Care**

Jazyk vypracování:

čeština

Zásady pro vypracování:

Vytvořte uživatelské prostředí pro ovládání provozně technických funkcí v Smart Home Care, pomocí sběrnice systému KNX. Provozně technické funkce (světlo, žaluzie, topení) bude možno ovládat pomocí desktopové aplikace z klasického PC nebo notebooku. Zároveň bude možné ovládat provozně technické funkce hlasem pomocí vybrané technologie.

1. Porovnejte a popište sběrnice systémy, používané v Smart Home Care.
2. Zhodnoťte současné způsoby provedení vizualizace provozně technických funkcí v Smart Home Care.
3. Vytvořte uživatelské prostředí pro ovládání provozně technických funkcí v inteligentní budově s asistivní péčí pomocí sběrnice systému KNX s ohledem na potřeby seniorů.
4. Realizujte ovládání provozně technických funkcí pomocí desktopového rozhraní přizpůsobeného pro klasické PC i nebo notebook.
5. Začleňte do vytvořeného prostředí možnost hlasového ovládání provozně technických funkcí pomocí technologie KNX v Smart Home Care.
6. Proveďte test funkčnosti a spolehlivosti implementovaného systému.

Seznam doporučené odborné literatury:

- [1] M. Valeš, Inteligentní dům, ERA group spol.s r.o., Brno, 2006, ISBN:80-7366- 062-8
- [2] Merz H., Hansemann T., Hubner C.; Automatizované systémy budov - Sdělovací systémy KNX/EIB, LON a BACnet, Grada Publishing, Praha 2007, ISBN 978-80-247-2367-9
- [3] Harper R.; Inside the Smart Home. Springer, London, 2003, ISBN1-85233-688-9

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Jan Vaňuš, Ph.D.**

Datum zadání: 01.09.2014

Datum odevzdání: 29.04.2016



---

doc. Dr. Ing. Eduard Sojka  
vedoucí katedry



---

prof. RNDr. Václav Snášel, CSc.  
děkan fakulty

Prohlašuji, že jsem tuto diplomovou práci vypracovala samostatně. Uvedla jsem všechny literární  
prameny a publikace, ze kterých jsem čerpala.

V Ostravě 22.4.2016

.....  
Belina!

Na tomto místě bych ráda poděkovala Ing. Janu Vaňušovi, Ph.D. za rady a pomoc při tvorbě této diplomové práce.

## **Abstrakt**

Práce se zabývá implementací aplikace pro ovládání inteligentní budovy vybavené sběrníkovým systémem KNX. Obsahuje porovnání sběrníkových systémů LonWorks, Nikobus, CIB, BACnet, M-bus a KNX v inteligentních budovách. Část práce je věnována porovnání vizualizací provozně technických funkcí, tematice Smart Home Care a asistivním technologiím. Praktická část práce analyzuje implementaci desktopové aplikace typu Universal Windows Platform pro Windows 10. Popisuje hlasové ovládání, komunikaci aplikace se sběrní KNX, využití architektury MVVM a přizpůsobení aplikace pro mobilní telefony.

**Klíčová slova:** ovládání inteligentní budovy, hlasové ovládání, rozpoznání hlasu, univerzální aplikace pro Windows 10, KNX

## **Abstract**

The thesis concerns with the implementation of a management application for smart home equipped with KNX system. It compares LonWorks, Nikobus, CIB, BACnet, M-bus and KNX bus systems in smart homes. Part of the thesis deals with operational and technical functions visualization comparison, Smart Home Care topics and assistive technologies. The practical part of the thesis analysis the implementation of desktop application of the Universal Windows Platform type for Windows 10. It describes voice control, application communication with KNX bus, MVVM architecture usage and optimalization for mobile devices.

**Key Words:** smart home control, voice control, speech recognition, Universal Windows Platform, KNX

# Obsah

<b>Seznam použitých zkratk a symbolů</b>	<b>9</b>
<b>Seznam obrázků</b>	<b>10</b>
<b>Seznam tabulek</b>	<b>11</b>
<b>1 Úvod</b>	<b>13</b>
<b>2 Motivace</b>	<b>14</b>
2.1 Použitelné technologie . . . . .	14
2.2 Zvolená technologie . . . . .	15
<b>3 Smart Home Care</b>	<b>17</b>
3.1 Inteligentní dům a jeho historie . . . . .	17
3.2 Péče o seniory . . . . .	18
3.3 Asistivní technologie . . . . .	18
3.4 Budoucnost . . . . .	20
<b>4 Sběrníkové systémy v inteligentních budovách</b>	<b>21</b>
4.1 LonWorks . . . . .	21
4.2 Nikobus . . . . .	21
4.3 CIB . . . . .	22
4.4 BACnet . . . . .	22
4.5 M-bus . . . . .	22
4.6 KNX . . . . .	23
<b>5 Vizualizace provozně technických funkcí</b>	<b>26</b>
5.1 Tlačítkové ovladače s LCD displejem . . . . .	26
5.2 Dotykové panely . . . . .	26
5.3 Mobilní telefony, tablety a PC . . . . .	28
5.4 Smart TV . . . . .	30
5.5 Vizualizace a ovládání pro seniory . . . . .	30
<b>6 Implementace</b>	<b>33</b>
6.1 Popis aplikace . . . . .	33
6.2 Universal Windows Platfom . . . . .	35
6.3 MVVM . . . . .	36
6.4 Komunikace s KNX sběrníci . . . . .	40
6.5 Hlasové ovládání . . . . .	41

6.6	Přizpůsobení pro mobilní telefony . . . . .	49
<b>7</b>	<b>Testování</b>	<b>51</b>
7.1	Ovládání provozně technických funkcí . . . . .	51
7.2	Hlasové ovládání . . . . .	52
<b>8</b>	<b>Závěr</b>	<b>53</b>
	<b>Literatura</b>	<b>54</b>
	<b>Přílohy</b>	<b>58</b>
<b>A</b>	<b>Obsah přiloženého CD</b>	<b>58</b>
<b>B</b>	<b>Uživatelská příručka</b>	<b>59</b>
B.1	Popis aplikace . . . . .	59
B.2	Instalace aplikace . . . . .	59
B.3	Hlavní okno aplikace . . . . .	60
B.4	Konfigurace . . . . .	60
B.5	Ovládání . . . . .	62
B.6	Nastavení . . . . .	63
B.7	Hlasové ovládání . . . . .	63



## Seznam použitých zkratek a symbolů

ACHE	– Adaptive Control of Home Environments
API	– Application Programming Interface
BACnet	– Building Automation and Controls Network
BNF	– Backus–Naur Form
CIB	– Common Installation Bus
DPT	– Datapoint Type
DTMF	– Dual-tone multi-frequency
EHS	– European Home Systems Protocol
EIB	– European Installation Bus
EN	– Evropská norma
ETS	– Engineering Tool Software
HVAC	– Heating, Ventilation, Air Conditioning
IoT	– Internet of Things
ISO	– International Organization for Standardization
KNX	– Konnex Bus
LCD	– Liquid Crystal Display
LON	– Local Operating Network
M-Bus	– Meter Bus
MVVM	– Model View ViewModel
OS	– Operační systém
OSI	– Open Systems Interconnection
PC	– Personal Computer
SCADA	– Supervisory Control and Data Acquisition
SDK	– Software Development Kit
SRGS	– Speech Recognition Grammar Specification
UWP	– Universal Windows Platform
Win	– Windows
XAML	– Extensible Application Markup Language
XML	– Extensible Markup Language

## Seznam obrázků

1	Topologie KNX sběrnice systému [14]	24
2	Individuální adresa [25]	24
3	Skupinová adresa [25]	25
4	Struktura telegramu [29]	25
5	Tlačítkový ovladač [19]	26
6	Dotykový panel EST-2/W [30]	27
7	Dotykový panel DIVUS SUPERIO [31]	28
8	OpenRemote vizualizace [22]	29
9	LOXONE vizualizace [32]	30
10	Aplikace iNels pro Samsung Smart TV [20]	31
11	Senior program	31
12	Náhled ovládací obrazovky	33
13	Náhled konfigurace	34
14	Diagram komponent	35
15	MVVM architektura [28]	36
16	Databázový model	37
17	Sekvenční diagram popisující zpracování hlasového příkazu	48
18	Aplikace v mobilním zařízení	49
19	Úspěšnost rozpoznání hlasových příkazů	52
20	Spuštění instalace	59
21	Úspěšná instalace aplikace Hestia	60
22	Konfigurace aplikace	61
23	Příklad špatně vyplněné skupinové adresy	62
24	Obrazovka ovládání	62
25	Náhled obrazovky s nastavením	63
26	Stavy hlasového rozhraní	64
27	Upozornění při nepovolení přístupu k mikrofonu	64

## Seznam tabulek

1	Podíl na trhu s operačními systémy v srpnu 2015 . . . . .	14
2	Podíl na trhu s operačními systémy v únoru 2016 . . . . .	14
3	Navigační povely . . . . .	64
4	Příkazy pro ovládání budovy . . . . .	64
5	Hlasové příkazy pro nastavení aplikace . . . . .	65
6	Hlasové příkazy pro konfiguraci aplikace . . . . .	67

## Seznam výpisů zdrojového kódu

1	Implementaci rozhraní INotifyPropertyChanged . . . . .	38
2	Použití rozhraní ICommand . . . . .	38
3	Definování použitého ViewModelu . . . . .	38
4	Použití atributů ItemsSource a SelectedItem . . . . .	39
5	Implementace IValueConvertoru . . . . .	39
6	Použití třídy CategoryToVisibilityLightsConverter . . . . .	40
7	Zapnutí osvětlení . . . . .	40
8	Inicializace rozhraní pro hlasové ovládání . . . . .	41
9	Průběh inicializace . . . . .	42
10	Element grammar . . . . .	43
11	Ukázka jednoduchého pravidla . . . . .	44
12	Odkaz na pravidlo . . . . .	44
13	Pravidlo se sémantickými hodnotami . . . . .	44
14	Ukázka opakování výrazu . . . . .	45
15	Definování možností zadání číselného vstupu . . . . .	45
16	Pravidlo pro zadání skupinové adresy . . . . .	46
17	Odkaz na pravidlo AddressFormat a jeho sémantický výstup . . . . .	47
18	Definování VisualStateManageru . . . . .	50

# 1 Úvod

Popularita chytrých řešení pro domácnosti v posledních letech rychle stoupá. Ať už se jedná o jednotlivá zařízení, jakými jsou žárovky, zámky dveří, vypínače osvětlení, termostaty, detektory úniku plynu nebo vody, chytré ledničky a pračky, ale také o kompletní elektroinstalace.

Také se zvyšuje průměrný věk populace a gramotnost seniorů v oblasti používání informačních a komunikačních technologií. Mnohem častěji využívají k zjednodušení každodenního života moderních prostředků. Zvyšují se také jejich požadavky na pohodlí a zdravotní péči. S tím souvisí rozvoj asistivních technologií, které ulehčují život nejen seniorům, ale také zdravotně postiženým a dlouhodobě nemocným.

Jednou z forem asistivních technologií je ovládání provozně technických funkcí v budově pomocí aplikací v počítači nebo telefonu. Možnosti vývoje desktopové aplikace s podporou hlasového ovládání a také zvolené řešení a odůvodnění jeho výběru jsou popsány v kapitole 2.

Následující kapitola se věnuje problematice Smart Home Care, popisuje inteligentní budovy a jejich historii. Dále jsou zde popsány možné formy asistivních technologií.

Kapitola 4 porovnává sběrníkové systémy používané pro realizaci inteligentních elektroinstalací v budovách. Věnuje se několika hlavním sběrníkovým systémům, a to LonWorks, Nikobus, CIB, BACnet, M-bus a KNX.

Dále jsou v teoretické části práce popsány vizualizační nástroje pro ovládání provozně technických funkcí. Mohou jimi být dotykové panely, aplikace v mobilních telefonech nebo tabletech, chytré televizory. Tyto i další způsoby vizualizace jsou popsány v kapitole 5.

Druhá polovina práce, počínaje kapitolou 6, popisuje implementaci vlastního řešení pro ovládání funkcí v inteligentních budovách s podporou hlasového ovládání a s ohledem na potřeby seniorů. Kapitola popisuje technologii UWP použitou pro vývoj aplikace. Část je věnována architektonickému vzoru MVVM a realizaci komunikace se sběrníicí KNX. Jedna podkapitola popisuje detailněji implementaci hlasového ovládání. Také jsou zde zmíněny některé postupy pro přizpůsobení aplikace pro mobilní telefony

Poslední kapitola je věnována popisu testování funkčnosti aplikace a hlasového ovládání.

## 2 Motivace

Jak již bylo zmíněno v úvodní kapitole, narůstá poptávka po chytrých řešeních pro inteligentní budovy, proto se jedná o část trhu s velkým potenciálem i do budoucna.

Cílem práce je vytvořit aplikaci pro ovládání funkcí inteligentní budovy vybavené sběrnice-  
vým systémem KNX. Má se jednat o aplikaci spustitelnou na desktopových operačních systémech  
s podporou hlasového ovládání.

V zadání práce nebyla specifikována technologie, pomocí níž má být implementace realizo-  
vána. Naskytá se tedy několik možností, jak lze tvorbu aplikace pojmout a to jak z hlediska  
cílového operačního systému tak i z hlediska zvolené vývojové technologie.

Na lokálním trhu pro desktopová zařízení figuruje několik hlavních operačních systémů. V  
České republice, ale i celosvětově je nejpopulárnější platforma Windows. Velmi malé procento z  
celkového podílu zaujímá operační systém OS X a Linux. V srpnu 2015<sup>1</sup> bylo nejvyšší zastoupení  
systému Windows 7. Windows 10, který byl oficiálně vydán v srpnu 2015, měl již více než  
osmi procentní podíl na trhu s operačními systémy. V únoru 2016<sup>2</sup> se druhým nejoblíbenějším  
systémem stal Windows 10 se skoro dvaceti procenty na trhu.

Tabulka 1: Podíl na trhu s operačními systémy v srpnu 2015

Operační systém	Win 7	Win 8.1	Win XP	Win 10	OS X	Ostatní
Zastoupení na trhu	52.57%	15.76%	9.63%	8.31%	3.85%	9.88%

Tabulka 2: Podíl na trhu s operačními systémy v únoru 2016

Operační systém	Win 7	Win 10	Win 8.1	Win XP	OS X	Ostatní
Zastoupení na trhu	46.19%	19.91%	12.76%	7.38%	4.57%	9.19%

Z důvodu vysokého zastoupení systému Windows na trhu s operačními systémy je také  
vývojářsky neatraktivnější. Existuje velké množství platforem, na kterých se dá vyvíjet pro  
tento operační systém (Java, Python, C++), ale nejvíce se s tvorbou aplikace pro Windows pojí  
platforma .Net a programovací jazyk C#.

### 2.1 Použitelné technologie

V následující odstavce se věnují technologiím, které lze pro implementaci zadání použít. V ka-  
pitole 5.3 je popsáno několik již existujících řešení aplikací pro ovládání inteligentních budov z  
PC, tabletu nebo mobilních telefonů.

<sup>1</sup><http://gs.statcounter.com/#desktop-os-CZ-monthly-201508-201508-bar>

<sup>2</sup><http://gs.statcounter.com/#desktop-os-CZ-monthly-201602-201602-bar>

### 2.1.1 Java

Jednou z možných platforem použitelnou pro tvorbu aplikace je jazyk Java. Výhodou tohoto řešení by byla kompatibilita napříč operačními systémy. V tomto případě je k dispozici Java Speech API <sup>3</sup>, které by bylo možné využít pro implementaci hlasového ovládání. Také je k dispozici open source knihovna pro zjednodušení komunikaci s KNX sběrnici. Jedná se o knihovnu Calimero <sup>4</sup>.

### 2.1.2 Python

Další možností je využít skriptovací jazyk Python, který má také k dispozici knihovnu pro rozpoznání řeči a to SpeechRecognition 3.3.3 <sup>5</sup>. Pro komunikaci s KNX sběrnici lze použít knihovnu knx 0.1.3 <sup>6</sup>.

### 2.1.3 .Net framework

V případě využití .Net frameworku lze použít několik programovacích jazyků, nejpoužívanějšími jsou C#, Visual Basic, Visual C++, F#. Pro rozpoznání hlasu lze využít Microsoft Speech Platform <sup>7</sup> a pro komunikaci s KNX sběrnici je k dispozici oficiální knihovna asociace KNX - Falcon SDK <sup>8</sup>.

### 2.1.4 C++

V jazyce C++ je k dispozici knihovna Voce<sup>9</sup> pro rozpoznání hlasu a jeho převodu na text. Je k dispozici také knihovna pro komunikaci se sběrnici KNX<sup>10</sup>.

## 2.2 Zvolená technologie

S použitím .Net frameworku je k dispozici několik typů aplikací, které lze v prostředí Windows vytvořit. Jsou jimi Windows Forms, Windows Presentation Foundation, Windows Store aplikace a Windows Universal Platform. Aplikace typu Windows Forms a Windows Presentation Foundation lze spustit na kterékoli z výše uvedených verzí systému Windows s kompatibilní verzí .Net Frameworku. Windows Store aplikace jsou vytvářeny pro operační systém Windows 8 a 8.1 a není je možné spustit na jiných verzích systému.

S Windows 10 byla představena platforma Universal Windows Platform, která umožňuje vytvářet aplikace kompatibilní se všemi verzemi systému Windows 10. Aplikace jsou spustitelné

---

<sup>3</sup><http://www.oracle.com/technetwork/java/jsapifaq-135248.html>

<sup>4</sup><http://calimero-project.github.io/>

<sup>5</sup><https://pypi.python.org/pypi/SpeechRecognition/>

<sup>6</sup><https://pypi.python.org/pypi/knx/0.1.3>

<sup>7</sup>[https://msdn.microsoft.com/en-us/library/hh361572\(v=office.14\).aspx](https://msdn.microsoft.com/en-us/library/hh361572(v=office.14).aspx)

<sup>8</sup><http://www.knx.org/ie/software/falcon/about/index.php>

<sup>9</sup><http://voce.sourceforge.net/>

<sup>10</sup><http://knx.nagilo.de/>

na desktopových zařízeních, mobilních telefonech, tabletech nebo zařízeních IoT. Na nižších verzích systému nelze spustit tento typ univerzální aplikace. Windows 10 by měl být již konečnou, poslední verzí systému Windows, která by měla být dále již jen aktualizována. Dle statistických údajů výše lze odvodit, že poptávka po systému Windows 10 bude nadále stoupat a to i z důvodu, že přechod na tuto verzi systému je nabízen zdarma. Proto je vývoj aplikací pro Windows 10 zajímavý.

Rozhodla jsem se pro vývoj Universal Windows Platform z několika důvodů. S vývojem pro operační systém Windows s použitím .Net frameworku a jazyku C# jsem nejlépe obeznámena. Windows Universal Platform je nová zajímavá technologie. Výslednou aplikaci lze spustit na mnoha různých zařízeních.



### 3 Smart Home Care

Se zlepšujícími se a dostupnějšími technologiemi se zvyšují nároky na zdravotní péči jak v nemocnicích, tak i v domácnostech. Rostou nároky populace na komfort bydlení a s tím je spojena zvýšená poptávka po automatizovaných řešeních.

Postoj seniorů k využívání moderních technologií v každodenním životě se postupně zlepšuje. Senioři začínají být více otevření využívání moderních zařízení za účelem zvýšení jejich pohodlí a bezpečí. Délka života se prodlužuje, a proto se zde pro příští desetiletí otevírají široké možnosti na trhu s inteligentními domácnostmi zaměřenými na péči o seniory.

Z dotazníku v diplomové práci Ing. Malinové [7] si lze vytvořit pohled na postoj seniorů k novým technologiím. Z 98 respondentů odpovědělo 72 ano na otázku, jestli by využívali nové technologie pro větší pohodlí a bezpečí. 70 osob by uvítalo domácí monitorování zdravotního stavu a odesílání naměřených hodnot lékaři. 45 by však stále chtělo ovládat domácnost manuálně a pouze 16 z dotazovaných by chtělo žít v inteligentní budově.

#### 3.1 Inteligentní dům a jeho historie

Za inteligentní dům je považována dům, který nejen že využívá moderní technologie, ale kde tyto technologie navzájem komunikují za účelem zvýšení komfortu obyvatel domu. Taková budova dokáže adaptivně reagovat na akce obyvatel, předvídat jejich chování a přizpůsobit se tak individuálním potřebám. Cílem automatizace je poskytnout obyvatelům celkovou kontrolu nad domem a to jak uvnitř budovy, tak i vzdáleně. Automatizační řešení zjednodušují každodenní život nejen starší populaci a handicapovaných.

Vše začalo elektrifikací a postupným začleňováním elektrických spotřebičů do domácností mezi lety 1915 a 1920. Ve 40. letech 20. století se množství domácností se zavedenou elektrickou energií mnohonásobně zvýšilo. Po druhé světové válce se na trhu s domácími spotřebiči začali objevovat myčky nádobí, elektrické vařiče. V 70. letech se běžnými součástmi stávaly toustovače, fény na vlasy, žehličky, vysavače, ale také centrální vytápění a termostaty. Později se začali hojně využívat televizory, DVD přehrávače. S postupným vývojem technologií se rozšířili možnosti využití jednotlivých spotřebičů a jejich automatizace. Nové přístroje lidem ušetřili čas a zjednodušili jejich využívání, ale stále je bylo nutné obsluhovat. Z těchto důvodů se rozšířila automatizace a spotřebiče se začali stávat inteligentními [1].

V roce 1966 Jim Sutherland vyvinul domácí počítač známý jako „Electronic Computing Home Operator“ neboli ECHO IV. Jednalo se o ručně vyrobený domácí automatizovaný systém zasazený v dřevěném kabinetu, který sloužil k uchovávání receptů, výpočtům nákupních seznamů. ECHO také ovládal teplotu v domě. Byl nabízen za \$10,000, ale neprodal se jediný kus [2].

Velkým průlomem v automatizaci bylo vyvinutí komunikačního protokolu X10 v roce 1975 firmou Pico Electronics ve Skotsku. Tento protokol slouží k dálkovému ovládání zařízení a osvětlení skrze elektrické vedení pomocí radiového signálu, který reprezentuje digitální informace

[4]. V roce 1984 byl poprvé použit termín „inteligentní dům“ skupinou National Association of Homebuilders ve Spojených státech Amerických.

V 90. letech 20. století se začali inteligentní domy dostávat více a více do povědomí široké veřejnosti. K tomu přispěl také dokumentární seriál britské televize BBC nazvaný Dream House, který po dobu 6 týdnů sledoval rodinu žijící v chytré domácnosti a ukazoval tak možnosti, které toto bydlení poskytuje.

V současnosti je možné centrálně ovládat okna i dveře. Je možné pomocí telefonního hovoru vypínat a zapínat jednotlivé spotřebiče. Pomocí pohybových senzorů lze automaticky zhasínat a rozsvěcovat světla v místnostech. Vnitřní osvětlení dokáže automaticky reagovat na vnější zdroje světla a regulovat svoji intenzitu. Stejným způsobem mohou fungovat i žaluzie. Je možné využívat automatizované systémy pro zavlažování rostlin. Dále systémy chytrých domácností nabízejí uživatelům spousty statistických údajů jako například informace o spotřebě elektřiny, vody, tepla apod. [3].

### 3.2 Péče o seniory

Procentuální zastoupení seniorů v populaci se stále zvyšuje, zároveň se lidé dožívají vyššího věku. S tím jsou spojeny také vyšší nároky na zdravotnický systém, a to jak na kapacity jednotlivých zařízení, tak na jejich komfort. Mnoho seniorů nerado opouští své domovy, nebo je opustit ze zdravotních důvodů nemohou.

Od roku 1970 se úmrtnost populace nad 80 let stále snižuje. V 50. letech 20. století byla pravděpodobnost dožití se věku od 80 do 90 let u žen průměrně 15-16% a u mužů průměrně 12%. V roce 2002 se tyto hodnoty zvýšili na 37% a 25%. V Japonsku, které je zemí s nejvyšší pravděpodobností dožití vysokého věku, je u žen hodnota této pravděpodobnosti 50%.

Zvyšuje se výskyt chronických onemocnění jako je cukrovka, srdeční onemocnění, artritida. Stoupá také počet onemocnění rakovinou. Převážně se jedná o rakovinu prostaty u mužů a rakovinu plic a prsu u žen [5].

### 3.3 Asistivní technologie

V péči o seniory a nemocné lze využít asistivních technologií. Jedná se o jakékoli zařízení, napomáhající plnit úkony, které by bez jeho použití nebylo pro uživatele možné vykonat. Asistivní technologie také zjednodušují či zvyšují bezpečnost při plnění těchto úkonů. Tato definice zahrnuje jak vybavení, tak služby poskytované v různých oborech.

Dle článku [6], který se zabývá přijetím asistivních technologií starší populací, by většina dotázaných respondentů využila asistivní technologie ke svému každodennímu životu a to především ke zlepšení pohodlí a zvýšení nezávislosti, která je pro ně důležitá.

### **3.3.1 Video monitorování**

Tato technologie se využívá především k dálkovému dohledu v bezpečnostním odvětví. Lze ji však využít také v domácích podmínkách k audio - video komunikaci prostřednictvím obrazovek na stěnách nebo pomocí televizoru. V závislosti na počtu kamer umístěných v budově, lze zjistit kde se osoba, kterou je nutné monitorovat, nachází. Lze tak sledovat, jestli nedošlo k poranění, nebo jestli pacient nespádl.

### **3.3.2 Monitorování životních funkcí**

Tento způsob dohledu nad pacientem lze realizovat pomocí náramku na zápěstí, který nepřetržitě sleduje puls, tělesnou teplotu a také pohyb pacienta. Několik dní je pacient monitorován a poté je každá odchylka zaznamenávána. Může dojít až k automatickému nouzovému volání na tísňovou linku v případě vyhodnocení systémem, že pacient omdlel, zkolaboval, nebo nastala jiná neobvyklá situace.

### **3.3.3 Detekce pádu**

Při této asistivní technologii je využíváno malé zařízení, které je umístěno v oblasti pasu nebo horní části hrudi pacienta. Pomocí akcelerometru a senzoru náklonu se vyhodnocuje několika různými způsoby, zda pacient nespádl. Jestliže se pacient nachází v horizontální poloze, zapne se alarm a po 15s varování. Během 15s zařízení pípá a může být vypnuto přemístěním do vertikální polohy [8]. Pro detekci pádu lze využít také mobilní telefony opatřené speciální aplikací. Pro operační systémy Android je kd dispozici aplikaci iFall [9]. V tomto případě je však nezbytné, aby pacient měl telefon stále u sebe.

### **3.3.4 Chrániče kyčlí**

Jedná se o speciální spodní prádlo z polypropylenu, která v případě pádu na kyčelní kloub zamezí jeho zlomení. Avšak používání těchto chráničů není příliš oblíbené z důvodu nízkého pohodlí během nošení.

### **3.3.5 Chytrá kuchyně**

V případě asistivních technologií se nemusí jednat pouze o jednotlivá zařízení, ale i o celý systém. Takovému systému se věnuje článek [10]. Je zde využita primární jednotka, která zpracovává informace ze spotřebičů a pomocí koncového zařízení např. televizor poskytuje uživateli přehled o jednotlivých zařízeních a zobrazuje různá varování. Dále detekuje nouzové situace, na které adekvátně reaguje. Veškerá data analyzuje pro potřeby obyvatel domu.

### 3.3.6 Mobilní aplikace a nositelná zařízení

V případě aplikace iBalance lze využít mobilní telefon pro rehabilitaci poruch rovnováhy, která může být způsobena stářím, nebo úrazem a zhoršením funkčnosti motorického centra. Aplikace využívá gyroskopu, akcelerometru a magnetometru pro snímání pohybu pacienta a zlepšování jeho motorických schopností [23].

### 3.3.7 Inteligentní domy s asistivní péčí

Propojením zmíněných technologií lze vytvořit inteligentní budovu pro monitorování seniorů s motorickým, zrakovým, sluchovým či jiným handicapem. Takovéto budovy musí splňovat určitá kritéria. Jedním z nich je vysoká přijatelnost. Například zařízení, u kterého je vyžadováno nošení speciálního vybavení, může být starší generaci na obtíž. Dalším důležitým aspektem je adaptace na změny situací, nebo schopností jednotlivých obyvatel, a zajištění tak jejich potřeb. Důležitá je také použitelnost. Služba musí být přístupná a nenáročná na ovládání.

Bylo vyvinuto mnoho budov, které jsou schopny rozpoznat aktivity obyvatel a zefektivnit tak asistivní péči. Takovéto systémy jsou schopny rozpoznat například chůzi, používání telefonu, konzumaci jídla. Pro rozpoznání těchto aktivit se využívají různé senzory jako akcelerometry, gyroskopy, mikrofony a kamery, které jsou umístěny v náramku na zápěstí [24].

V USA vyvinuly adaptivní dům ACHE, který pomocí neuronové sítě ovládá vytápění a osvětlení bez předchozího naprogramování obyvateli. Nepřetržitě sleduje akce prováděné obyvateli a předvídá jejich další chování. Jeho hlavním účelem je snížit náklady na provoz budovy a zvýšit pohodlí obyvatel. V Japonsku je postaveno 13 „Welfare Techno Houses“, jejichž úkolem je zvýšit kvalitu života jak seniorů, tak i jejich opatrovníků. V Osace je chytrý dům Dr. Matsuoka, který pomocí 167 senzorů automaticky rozpozná neobvyklou událost způsobenou nemocí nebo nehodou [11].

Ve Velké Británii se nachází CareNet. Jedná se o asistivní dům s nouzovým alarmem a ambulantním monitorováním. Je opatřen velkým množstvím senzorů a to i nositelných. Monitoruje velké množství údajů jako tělesnou teplotu, puls, EKG. Pro zpracování těchto údajů slouží centrální jednotka. Jako případová studie zde bylo využito monitorování pacienta po operaci mozku.

## 3.4 Budoucnost

Neustálý vývoj v oblasti informačních technologií, elektroniky a komunikační techniky otevírá nové možnosti v oblasti inteligentních budov s asistivní péčí. Zvyšuje se dostupnost jednotlivých zařízení a snižují se náklady na jejich výrobu. Spousta řešení v této oblasti jsou experimentálního rázu, a proto je zde velký potenciál pro další výzkum a vývoj. Je potřeba lépe zmapovat potřeby seniorů a co nejlépe jim systémy přizpůsobit. Další vývoj je také nutný v oblasti bezpečnosti chytrých systémů, jejich spolehlivosti a účinnosti.

## 4 Sběrníkové systémy v inteligentních budovách

Sběrníková topologie je způsob propojení jednotlivých prvků uvnitř budov pomocí sběrnice, která představuje hlavní přenosový kanál. Sběrnice slouží k přenosu informací uvnitř celé infrastruktury. Pro přenos informací se využívají různá řešení. Aby se tato řešení sjednotila a bylo tak možné propojovat zařízení různých výrobců, vznikly jednotné systémy sběrnice, jako například KNX, LON.

### 4.1 LonWorks

LonWorks je otevřený decentralizovaný sběrníkový systém, který je standardizovaný normou EN 14908. Byl vyvinut firmou Echelon v letech 1989-1992. Pro přenos informací se využívá lokální operační síť, této technice se říká LON (Local Operating Network).

Pro komunikaci po sběrnici se využívá protokol LonTalk, který byl vyvinut v 90. letech jako univerzální a levné komunikační spojení. Jako přenosové médium se využívají kroucené páry vodičů, elektro rozvodná síť, vysokofrekvenční rádiové vlny, infračervené spojení, koaxiální kabel a optická vlákna. Topologie sítě může být libovolná. Hierarchii LON lze rozdělit na domény. Ty mohou být složeny až z 255 podsítí po 127 uzlech. Přenosové rychlosti v této instalaci se pohybují od 600b/s až 1,25Mb/s dle použitého přenosového média a délky spojení.

Technologie LonWorks je využívána v automatizaci budov (výtahy, klimatizace, topení, osvětlení), řízení domácích spotřebičů, měření, regulaci a v mnoha dalších oblastech. Důvodem využití této technologie je vysoká spolehlivost a zabezpečení sítě. Využívá speciální autentizační algoritmus, je vysoce flexibilní a má nízké instalační nároky [15] [16].

### 4.2 Nikobus

Nikobus je sběrníkový systém navržený belgickou firmou Niko, který pro přenos informací využívá sběrníkové vedení s bezpečně malým napětím 9V. Jedná se o hybridní (částečně decentralizovaný) systém, který byl vyvinut speciálně pro domy a byty.

Přes sběrnici lze na jednu řídicí jednotku připojit maximálně 256 senzorů. Jejich naprogramování nevyžaduje vždy připojení k počítači, lze jej však pomocí zařízení PC-LINK výhodněji a přehledněji parametrizovat pomocí počítače. Systém nepracuje s žádnými složitými příkazy, posílají se pouze povely zapnout/vypnout.

Topologie systému může být libovolná, dodatečné rozšíření instalace je jednoduché. Jako přenosové médium se využívá kroucená dvojlinka. Sběrnice je napájena z řídicí jednotky, ty lze rozdělit na tři základní typy - spínací, stmívací a žaluziová jednotka [17].

### 4.3 CIB

CIB je sběrnice používaná v inteligentní elektroinstalaci iNels. Ta vznikla ze spolupráce firem Teco, a. s. a Elko EP s. r. o.. Jedná se o dvoudrátovou sběrnici sdružující rychlou komunikaci a napájení senzorů a akčních členů.

Sběrnice je jednoduchá na instalaci, akční členy a senzory se propojují dvouvodičovými kabelem. Topologie systému může být libovolná kromě kruhového zapojení. Napájení a data jsou vedena společně ve dvou vodičích a minimalizuje se tak počet vodičů v instalaci.

Komunikace na sběrnici je realizována modelem master-slave. Na jednu větev je možné připojit až 32 jednotek. Centrální jednotku lze rozšířit v případě potřeby o další větve pomocí externích modulů master, které obsahují další dvě větve. Toho lze využít nejen pro zvýšení počtu připojených akčních členů a senzorů, ale také pro zvětšení rozlehlosti systému. Jednotlivé moduly master lze pomocí metalického kabelu umístit do vzdálenosti 300m od řídicí jednotky, nebo s použitím optického kabelu do vzdálenosti 1,7km [18].

### 4.4 BACnet

BACnet je standardizovaný protokol speciálně vyvinutý pro komunikaci zařízení mezi systémy automatizace budov. Samotný není klasickou komunikační sběrnici. Byl vytvořen za účelem integrace zařízení od různých výrobců. V roce 2003 se stal evropským standardem v rámci CEN označeným ISO 16484-5.

BACnet specifikuje tři části. Skládá se z objektů, služeb pracujících s objekty a protokolu síťové a fyzické vrstvy. Byl navržen tak, že zahrnuje nejvíce využívané funkce pro monitorování a řízení systémů. Jsou jimi analogové hodiny, dvoustavové vstupy a výstupy, ovládání a plánování [12] [13].

BACnet je využíván především v následujících oblastech:

- řízení HVAC (Heating, Ventilation, Air Conditioning)
- detekce a hlášení požárů
- řízení osvětlení
- zabezpečovací systémy
- inteligentní výtahy
- řídicí rozhraní zařízení

### 4.5 M-bus

Evropský standard M-bus je průmyslová datová sběrnice určená k přenosu dat a řízení. Byla vyvinuta za účelem měření a regulace. Přenos se provádí po dvouvodičové sběrnici, přičemž

účastnické stanice mohou být přímo napájeny. Prvotním účelem byl sběr dat z měřičů odběru pitné vody, plynu, elektrické energie apod.

Vzhledem k účelu sběrnice jsou na ni kladeny specifické nároky na množství připojených zařízení, těch může být řádově několik set, a vysokou vzdálenost až několika kilometrů. Jsou zde však nízké nároky na časté odečítání naměřených hodnot a tedy nízké nároky na odezvu.

Komunikace po sběrnici je realizována na principu master-slave. Přenosová rychlost na sběrnici se pohybuje mezi 300 a 9600 Bd. Datové bloky jsou zabezpečeny pomocí kontrolního součtu.

## 4.6 KNX

KNX je evropská instalační sběrnice vycházející ze tří technologií sběrnic - EIB, BatiBUS a EHS, která slouží ke komunikaci akčních členů a senzorů uvnitř budov. V roce 2003 byla sběrnice KNX zařazena do evropské normy EN 50090.

Jedná se o decentralizovaný sběrnice systém, tvořený akčními členy (topení, žaluzie, osvětlení, ...) a senzory (teplotní čidlo, tlačítka, ...). Jednotlivé akční členy nejsou přímo spojeny se senzory, ale komunikují spolu za pomoci sběrnice pracující s napětím 28V.

KNX systém se skládá z následujících komponent:

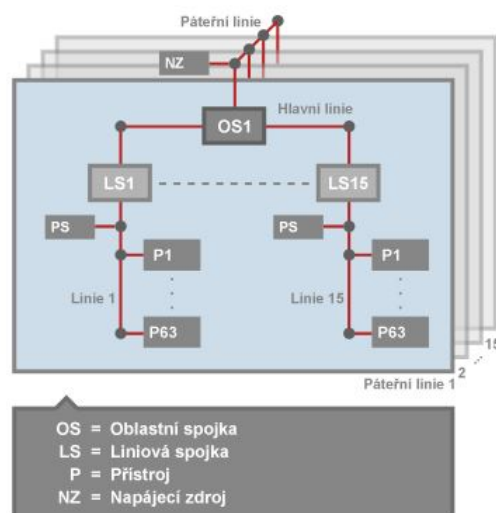
- Napájecí zdroj
- Snímače (termostaty, tlačítka)
- Akční členy (žaluzie, spínací relé)
- Sběrnice

Přenos informací pomocí sběrnice KNX je možné realizovat s využitím různých přenosových médií. Nejpoužívanější je kroucený pár (KNX.TP), dále se využívá klasická silová síť (KNX.PL), rádiový přenos (KNX.RF), infračervený přenos (KNX.IR) a Ethernet/IP (KNX.IP). Jednotlivé prvky připojené ke sběrnici mají svou unikátní adresu a komunikují spolu pomocí telegramů. Programování prvků je realizováno pomocí softwaru ETS, ten je však finančně nákladný [12] [13].

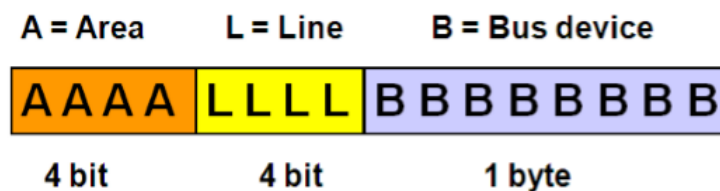
KNX topologii lze rozdělit do několika částí. První z nich je linie, která se skládá maximálně ze 4 liniových segmentů o maximálním množství 64 přístrojů. V případě použití více linií je nutné tyto linie propojit liniovými spojkami, tímto způsobem lze spojit až 15 linií na jednu hlavní a vytvořit tak oblast. Pomocí páteřní linie lze propojit několik oblastí.

### 4.6.1 Individuální adresa

K jednoznačné identifikace přístroje na sběrnici slouží individuální adresa, která popisuje také umístění přístroje v rámci topologie. Individuální adresa je tvořena posloupností 16bitů, kde první 4bity představují oblast, ve které se přístroj nachází. Další 4bity reprezentují linií a posledních 8 bitů je určeno pro adresování zařízení v rámci linie.



Obrázek 1: Topologie KNX sběrnicevého systému [14]



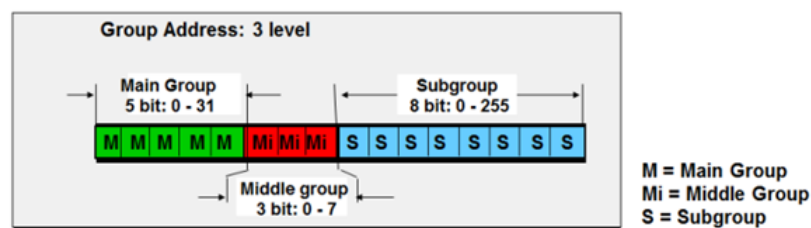
Obrázek 2: Individuální adresa [25]

S individuální adresou se při programování v softwaru ETS pracuje v desítkové soustavě. Pro adresování páteřní linie je vyhrazena 0 v první části adresy, pro adresování hlavní linie je vyhrazena 0 ve druhé části. Adresa 1.1.15 představuje patnácté zařízení v první oblasti a první linii. Individuální adresa se dále využívá pro diagnostiku sběrnice, pro změny instalace přeprogramováním zařízení. V běžném provozu však nemá význam.

#### 4.6.2 Skupinová adresa

Skupinová adresa se v instalaci vyskytuje vždy alespoň dvakrát a to u snímače a akčního členu. Tímto způsobem se přístroje propojí. Snímač odešle skupinou adresu a akční člen ji vyslechne a provede odpovídající akci. Skupinová adresa může být dvouúrovňová, tříúrovňová nebo volně definovatelná. Nejčastěji se využívá tříúrovňová varianta skupinových adres. Jednotlivé úrovně (hlavní, střední, podskupina) jsou odděleny lomítky. Hlavní skupina může obsahovat 16 adres, střední 8 a podskupina až 256 adres, celkově je tedy možné využít 32 768 adres. Příkladem tříúrovňové adresy je adresa 2/8/25. Skupinové adresy mohou být zařízením přiřazovány podle potřeby nezávisle na jejich umístění v topologii [25].



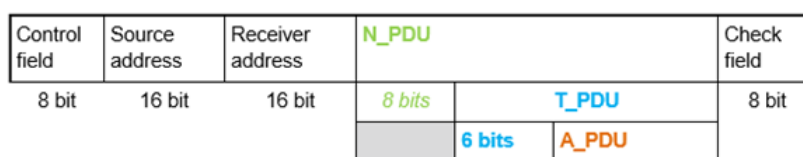


Obrázek 3: Skupinová adresa [25]

#### 4.6.3 Telegram

Telegram slouží pro přenos informací po sběrnici. Vyměňují si je jednotlivá KNX zařízení, a to ve formě datových telegramů a zpětného hlášení (potvrzovací telegram). Tento provoz je založen na referenčním modelu OSI podle normy ISO 7498. Telegram je tvořen posloupností bitů. Nejkratší telegram má délku 9bytů a nejdelší až 23bytů. Telegram je rozdělen do následujících polí:

- kontrolní pole (8 bitů)
- adresa odesílatele (16 bitů)
- adresa příjemce (17 bitů)
- přepravní pole (3 bity)
- údaje o délce datového pole (4 bity)
- datové pole (1 až 16 bytů)
- zkušební pole (8 bitů)



Obrázek 4: Struktura telegramu [29]

Telegramy se přenášejí jako paket znaků UART. Na začátek paketu se vkládá startovací bit, ten má implicitní hodnotu 0. Paket pokračuje sekvencí 8 datových bitů. Dále je zaslán paritní bit a zakončen je koncovým bitem s implicitní hodnotou 0. Před odesláním dalšího znaku se za paket vkládá mezera dvou bitů.

Aby bylo zabráněno vzniku konfliktů v případě, že dvě zařízení chtějí vysílat současně telegram, využívá se přístupová metoda CSMA/CA. [13].

## 5 Vizualizace provozně technických funkcí

Jako vizualizace je označováno grafické zobrazení provozních stavů jednotlivých funkcí a jejich případné ovládání. V průmyslových instalacích se používá pojem SCADA (Supervisory Control And Data Acquisition). Tento pojem se používá pro označení softwaru, pomocí něhož lze zobrazovat a ovládat funkce systému.

### Současné trendy vizualizace ovládání inteligentních budov

- tlačítkové ovladače s LCD displejem
- dotykové panely
- mobilní telefony, tablety nebo PC
- Smart TV

Jednotlivé způsoby vizualizace jsou dále popsány a u každé varianty je vypsáno několik konkrétních příkladů, které ilustrují danou skupiny vizualizačních nástrojů.

#### 5.1 Tlačítkové ovladače s LCD displejem

Nejjednodušším způsobem vizualizace funkcí v budovách je pomocí malých LCD displejů, které jsou obohaceny o hardwarová tlačítka sloužící k ovládání funkcí. Tento způsob lze využít pro vizualizaci a ovládání například osvětlení, vytápění nebo klimatizace [19].



Obrázek 5: Tlačítkový ovladač [19]

#### 5.2 Dotykové panely

Dotykové panely lze využít pro vizualizaci a ovládání funkcí v jednotlivých místnostech, také jako centrální prvek celého objektu. Výhodou tohoto způsobu vizualizace je možnost vlastního

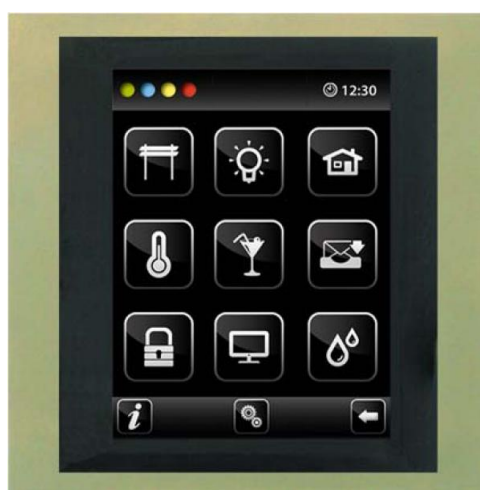
nastavení zobrazování. Často je možné konfigurovat rozložení jednotlivých obrazovek, co a kde se na obrazovkách bude zobrazovat. Je možné nastavovat velikost písma na tlačítkách, barevná schémata atd.

Oproti předešlému způsobu je tato vizualizace komfortnější a nabízí mnohem více možností.

### 5.2.1 Ovládací dotyková jednotka - EST-2/W

Jedná se o dotykovou jednotku pro elektroinstalaci iNels společnosti Elko. Dodává se ve variantě zapuštěné do zdi i ve variantě, která se umístí na zeď. Jednotka obsahuje 3,5" barevný displej o rozlišení 240x320.

Uživatel si může panel nastavit podle svých představ. Je možné obrazovku rozdělit na matici o 4, 6, 9 nebo 12 pozicích. Na každou pozici lze vybrat jednu z 30-ti nabízených ikon [30].

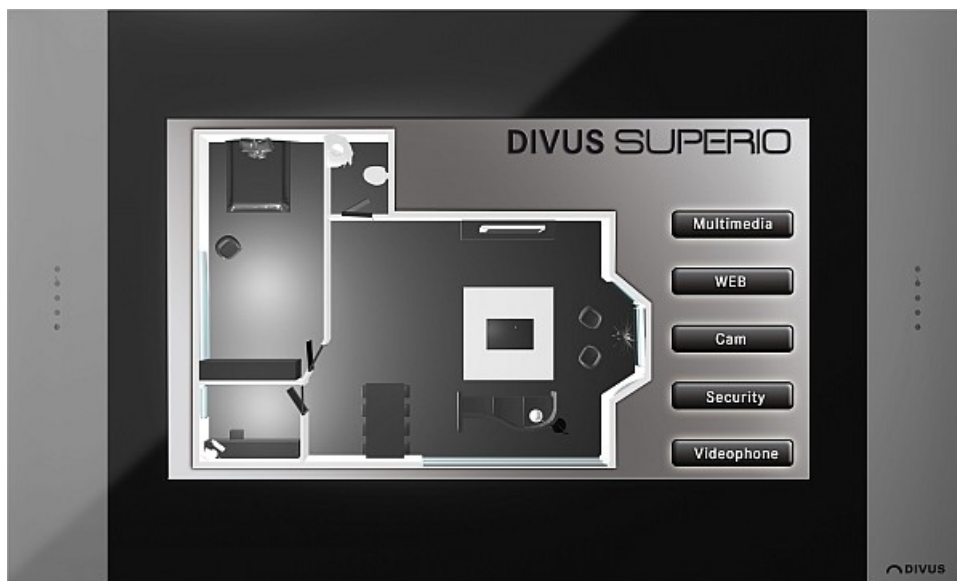


Obrázek 6: Dotykový panel EST-2/W [30]

### 5.2.2 Dotykové panely DIVUS

DIVUS je řada LED vizualizačních displejů, které lze použít v obytných budovách, kancelářích, restauracích a na dalších místech. Dotykový panel je určený k montáži přímo na stěnu místnosti. Tyto displeje je možné zakoupit v několika variantách.

DIVUS DIVA je 22" Full HD LED displej. Obsahuje vestavěnou televizi a ovládat ji lze pomocí multidotykové plochy. DIVUS SUPERIO je řada dotykových panelů o velikostech 10", 15" a 19". Tuto řadu panelů lze rozšířit o Bluetooth a bezdrátové rozhraní. DIVUS DOMUS je navržen ve velikostech 6,5", 10", 15" a 19". Tato varianta je určena pro odbornější veřejnost. Obsahuje OS Windows Embedded a lze jej snadno konfigurovat pro jednotlivé aplikace. DIVUS HOME lze zakoupit ve dvou variantách velikostí a to 10" a 15". Je velice tenký, nad úroveň stěny vyčnívá pouze 25mm dotykového panelu.



Obrázek 7: Dotykový panel DIVUS SUPERIO [31]

### 5.3 Mobilní telefony, tablety a PC

Tento způsob vizualizace a ovládání funkcí v inteligentních budovách je velice populární, především varianty v tabletech a mobilních telefonech. Je to nejkomfortnější způsob ovládání inteligentního domu.

Pro toto řešení existuje velké množství softwaru, který lze využít. Lze jej nainstalovat na zařízení s operačním systémem Android, existují také varianty pro iPhone i iPad a také pro webové prohlížeče. Některý software je dodáván přímo výrobcem inteligentní elektroinstalace. Na trhu jsou však i varianty, které nejsou dodávány výrobcem inteligentních instalací, tento software lze použít jen pro podporované řídicí prvky.

#### 5.3.1 KNX Dashboard

Je desktopová aplikace pro vizualizaci ovládání inteligentních budov pro operační systém Windows a to pro verze 8, 8.1 a Windows 10. Aplikace komunikuje se sběrnici KNX, buď pomocí KNX IP routeru, nebo za pomoci serveru KNXWeb2 [33].

#### 5.3.2 HomeGenie

HomeGenie je programovatelný server pro automatizaci a ovládání. Jedná se o platformu s otevřeným kódem. Podporuje tvorbu scénářů, hlasové ovládání. K serveru se přistupuje pomocí webového rozhraní, je tedy dostupný jak z počítačů, tak i z tabletů nebo mobilních telefonů. Pro mobilní zařízení jsou k dispozici aplikace pro práci se serverem. Uživatelské rozhraní je možné vytvořit dle vlastních požadavků. Lze přidávat ovládací prvky nebo zobrazovat statistická data [34].

### 5.3.3 OpenRemote

OpenRemote je volně dostupná softwarová platforma, která slouží k ovládání a vizualizaci stavů provozně technických funkcí v budovách. Je nezávislý na automatizačním protokolu a je možné jej spustit na mnoha zařízeních. Skládá se ze tří částí OpenRemote Designer, OpenRemote Controller a OpenRemote Panels.



Obrázek 8: OpenRemote vizualizace[22]

#### OpenRemote Designer

Jedná se o cloudovou aplikaci, která umožňuje návrh uživatelského prostředí pro ovládací panely. Je možné navrhovat rozhraní pro zařízení s operačním systémem Android, pro iPhone, iPod Touch i iPad. Je možné navrhovat uživatelská rozhraní i pro webové prohlížeče.

Uživatelské rozhraní lze vytvářet s různými funkcemi na míru každému uživateli a přiřadit jim odlišná práva pro ovládání.

#### OpenRemote Controller

OpenRemote Controller propojuje ovládací panely s uživatelským rozhraním. Také poskytuje množství příkazů přes několik protokolů do jediného makra, které může být spouštěno na základě plánu nebo akce.

#### OpenRemote Panels

Podporuje vykreslování ovládacích panelů v mobilních telefonech a tabletech s OS Android, na zařízeních iPhone, iPad a iPod Touch i v PC pomocí webového prohlížeče. Uživatelské rozhraní a scény lze upravovat pomocí OpenRemote Designer [22].

#### 5.3.4 LOXONE miniserver

Loxone miniserver je schopen z klasické instalace vytvořit elektroinstalaci inteligentní. Umožňuje ovládat osvětlení, žaluzie i vytápění domácnosti. Všechn software je dostupný zdarma. Jediné za co uživatel platí, je hardwarové vybavení.



Obrázek 9: LOXONE vizualizace [32]

Loxone miniserver lze použít v klasické elektroinstalaci, ale lze jej také využít v hybridním zapojení například s KNX prvky. Na trhu je ve dvou variantách. První je LOXONE miniserver, který se využívá v nových stavbách. Druhou variantou je LOXONE miniserver GO, který lze použít v již existujících budovách.

V případě použití LOXONE v klasické elektroinstalaci, je jeho zapojení velice jednoduché. Po zapojení je k dispozici konfigurační software, pomocí něhož lze jednoduchým způsobem vytvořit uživatelské rozhraní pro ovládání celé budovy [32].

#### 5.4 Smart TV

Domácnost lze ovládat také pomocí televizorů, musí se však jednat o chytré televize. Pro inteligentní elektroinstalaci iNels je možné do televize Samsung Smart TV nainstalovat aplikaci, která umožňuje ovládání celého domu. Toto řešení poskytuje možnost ovládat osvětlení, žaluzie a vytápění v jednotlivých místnostech pomocí ovladače a televizní obrazovky [20].

#### 5.5 Vizualizace a ovládání pro seniory

V oblasti vizualizace provozně technických funkcí určené pro seniory jsou velké mezery. Výrobků zacílených přímo na tuto skupinu je velice málo. Je však možnost využít spousty jiných vizua-



Obrázek 10: Aplikace iNels pro Samsung Smart TV [20]

lizačních systémů a upravit si je na míru. Například pomocí OpenRemote Designeru je možné vytvořit vizualizace i pro seniory. Totéž platí pro LOXONE miniserver.

### 5.5.1 InHome

InHome je systém vyvinutý firmou Insight Home, který umožňuje ovládání inteligentního domu. Lze pomocí něj ovládat osvětlení, žaluzie, vytápění, zabezpečení a mnoho dalších funkcí. V jeho vizualizační části, lze uzpůsobit ovládání do mnoha profilů s různými funkcemi. Speciální profil lze použít například pro děti, seniory a ostatní členy domácnosti.



Obrázek 11: Senior program [21]

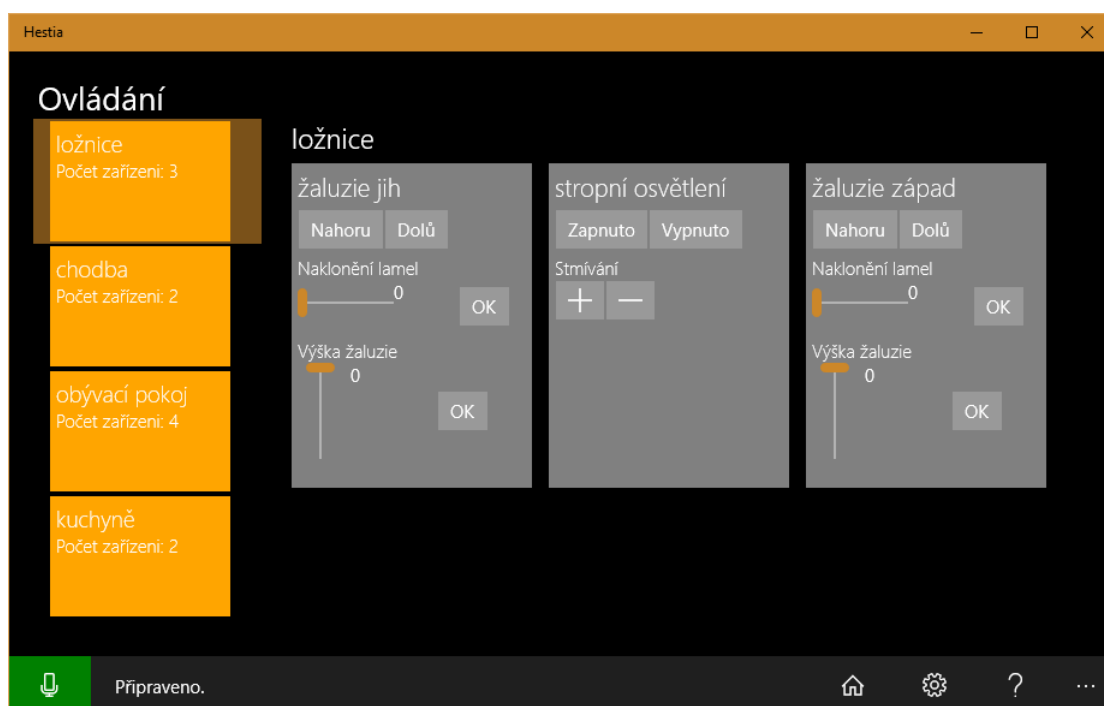
Senior program umožňuje seniorům jednoduché a pohodlné ovládání funkcí budovy. Zobrazí jim ovládací prvky na panelu s dvojnásobnou velikostí. Umožňuje jim ovládat pouze základní funkce, aby vše bylo co nejjednodušší. Je možné nastavit hlasové ovládání například světel. Umístit do pokoje komunikační prvek, který je propojen s mobilním telefonem příbuzného. Také lze pomocí zabudované kamery zkontrolovat, kdo se nachází před vstupními dveřmi a pomocí hlasového příkazu otevřít. [21]



## 6 Implementace

Většina výše zmíněných aplikací je určena pro mobilní platformu Android. Pro operační systém Windows 10 příliš aplikací neexistuje, proto jsem si zvolila tuto platformu jako cílovou. Dalším důvodem je použitelnost výsledné aplikace jak na desktopových zařízeních, tak na zařízeních mobilních. Zajímavou vlastností je možnost hlasového ovládání a přístupnost pro seniory. Při implementaci jsem se zaměřila na vývoj Universal Windows Platform. Využila jsem .Net Framework a jazyk C#. Dále jsem použila architektonický vzor MVVM, který je v textu stručně popsán.

### 6.1 Popis aplikace



Obrázek 12: Náhled ovládací obrazovky

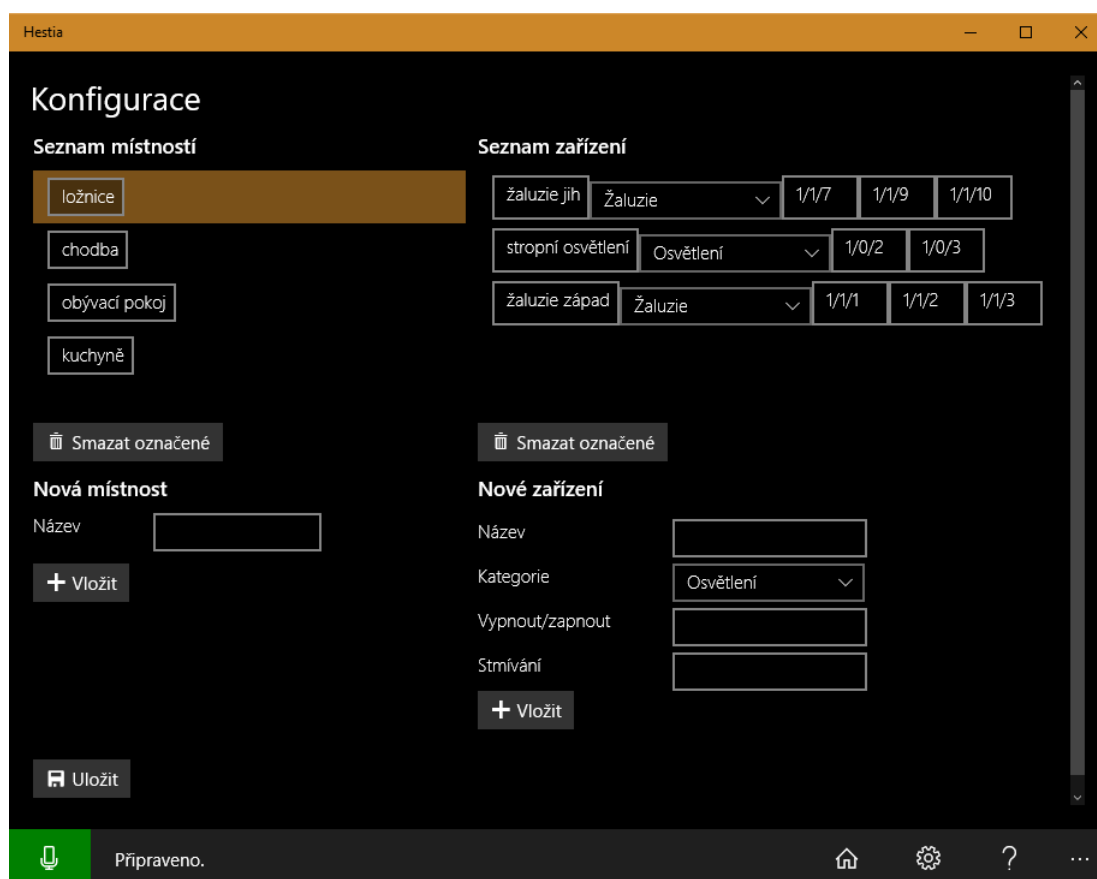
Aplikace Hestia <sup>11</sup> umožňuje ovládání inteligentní budovy vybavené sběrníkovým systémem KNX. Skládá se z několika obrazovek. Pro uživatele je tou nejdůležitější obrazovka s ovládacími prvky. Ta obsahuje seznam místností v jejichž detailech se zobrazují jednotlivá, předem nakonfigurovaná, zařízení viz obrázek 12. Na základě konfigurace, která bude popsána později, se vykreslí odpovídající ovládací prvky. V případě osvětlení je možno ovládat rozsvěcování a zhasínání světel. Je možné také měnit jas světel pomocí tlačítek pro stmívání. Pokud se jedná o zařízení typu žaluzie, pak je v vykreslen ovládací prvek, který může obsahovat několik funkcí.

<sup>11</sup>Hestia je v řecké mytologii bohyně architektury, posvátného ohně, rodinného krbu, rodiny a státu.

Prvním z nich je možnost stahování žaluzií nahoru a dolů. Dále je možné žaluzii stáhnout pouze do určité výšky a v poslední řadě naklopit lamely žaluzií do požadované polohy.

Další důležitou obrazovkou je nastavení aplikace. Zde má uživatel možnost upravit některé parametry aplikace pro přizpůsobení svým individuálním potřebám. Lze změnit jazyka aplikace, k dispozici je český a anglický jazyk. Dále je možný výběr ze dvou barevných schémat, světlého a tmavého. Posledním nastavením je změna velikosti písma. To usnadní práci s aplikací především seniorům. Na výběr jsou tři velikosti písma, malé, střední a velké.

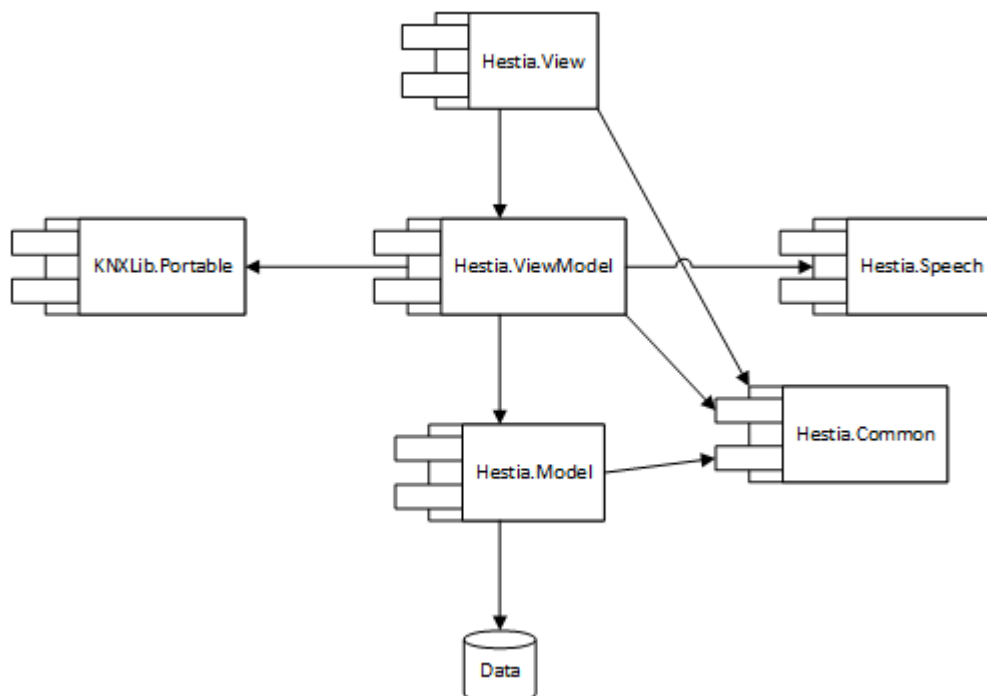
Předpokladem správné funkčnosti aplikace je konfigurace. Ta probíhá na zvláštní obrazovce, na kterou lze přistoupit z nastavení aplikace. Za účelem lepší přehlednosti je nejprve vytvořen seznam místností v budově. Poté je pro jednotlivé místnosti vytvořen seznam zařízení. Při tvorbě zařízení je vybrán typ zařízení a je zadána alespoň jedna skupinová adresa některé z možných funkcí. Konfiguraci by měla provádět osoba obeznamovaná s KNX sběrníkovým systémem a konkrétním naprogramováním sběrnice. Je nutné znát jednotlivé skupinové adresy a typy zařízení, které se v topologii nacházejí. Bez této znalosti není možné zajistit funkčnost ovládání.



Obrázek 13: Náhled konfigurace

Aplikace se skládá z několika komponent, těmi základními jsou Hestia.View, Hestia.Model a Hestia.ViewModel, které tvoří kostru MVVM architektonického vzoru. Ten je popsán dále v

textu. Další komponentou je KNXLib.Portable. Jedná se o volně dostupnou knihovnu pro komunikaci s KNX sběrnici. Hestia.Speech zajišťuje logiku hlasového rozpoznání a předává informace dále v aplikaci. Hestia.Common je sdílená komponenta obsahující globální proměnné a metody používané napříč aplikací.



Obrázek 14: Diagram komponent

## 6.2 Universal Windows Platform

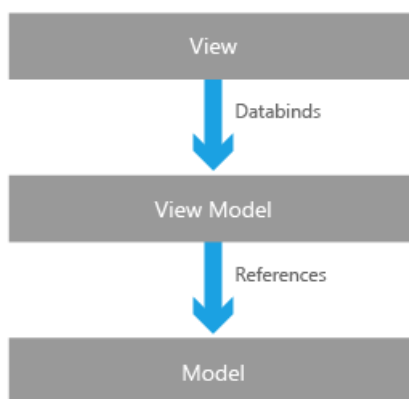
S Windows 8 byla představena nová architektura aplikací Windows Runtime (WinRT), která je také součástí systému Windows Phone 8.1. Umožnila vývoj univerzálních aplikací pro Windows 8 pomocí sdíleného kódu.

S příchodem Windows 10 byla představena Universal Windows Platform, jejíž jádro je založeno na WinRT a poskytuje tak společnou platformu dostupnou na všech zařízeních s Windows 10. Aplikace může využívat API společná pro všechna zařízení, ale také API která jsou specifická pro zařízení, na kterém aplikace běží. Stačí tak vytvořit jeden instalační balíček, který je možné spustit na kterémkoli zařízení s Windows 10.

Předpokladem pro vývoj univerzálních aplikací je operační systém Windows 10 a vývojové prostředí Visual Studio 2015 s balíčkem pro vývoj univerzálních aplikací Universal Windows App Development Tools. Dále je potřeba systém přepnout do vývojářského režimu, jinak nelze aplikaci na zařízení spustit a testovat. V případě použití operačních systémů Windows 7 nebo Windows 8.1 není možné aplikace spouštět lokálně, pouze za použití vzdálených zařízení s operačním systémem Windows 10 [27].

## 6.3 MVVM

MVVM je architektonický vzor, který slouží k oddělení uživatelského rozhraní od logiky aplikace. Skládá se ze tří logicky oddělených částí a to Model, View a ViewModel. Vrstva View představuje uživatelské rozhraní využívá se zde jazyk XAML. Model představuje vrstvu k přístupu k datům, se kterými aplikace pracuje. ViewModel spojuje Model a View, zachycuje aktuální stav aplikace a obstarává logiku jednotlivých View.



Obrázek 15: MVVM architektura [28]

### 6.3.1 Model

Jak již bylo zmíněno, Model představuje vrstvu přistupující k datům aplikace. V tomto případě je Model tvořen třídami `Room`, `Device`, `AdressType` a `FunctionType` a třídami pro jejich načtení z XML souboru.

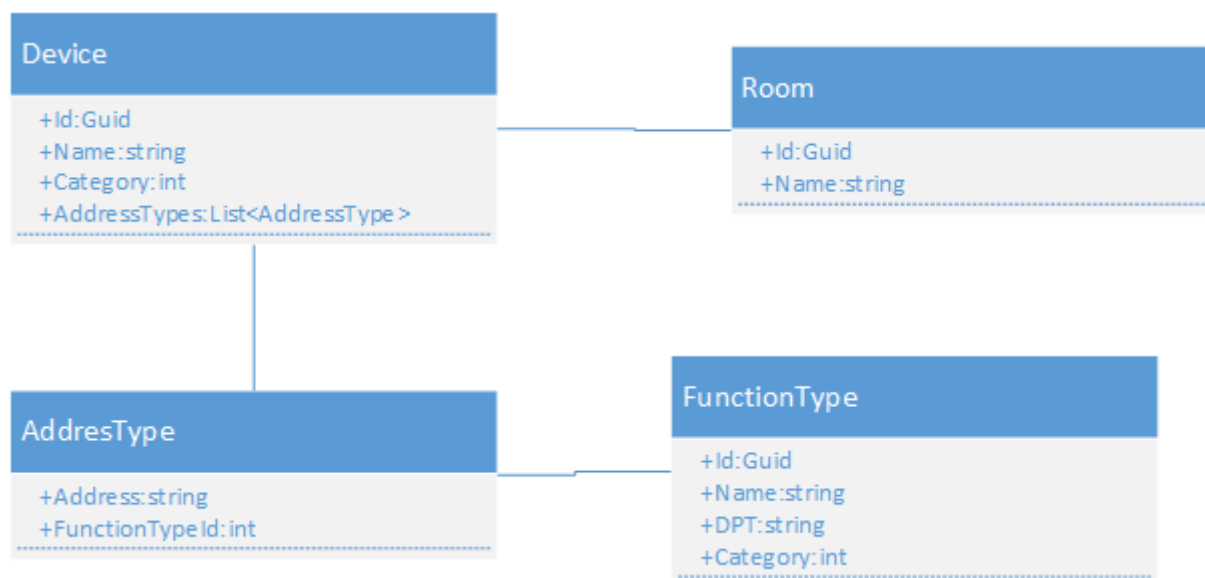
Pro ukládání celé struktury se využívá XML serializace. Před názvy tříd se vloží tag `[XmlRoot("NazevTridy")]` a před jednotlivé property buď `[XmlElement("NazevVlastnosti")]`, nebo `[XmlAttribute("NazevVlastnosti")]`. Po zavolání příslušné metody pro ukládání lze vytvořit XML soubor, který obsahuje jednotlivé objekty a jejich property ve tvaru xml tagů podle jejich definice v třídách.

Třída `Room` představuje místnosti v budově a obsahuje jméno, id a seznam jednotlivých zařízení.

Třída `Device` reprezentuje jednotlivá zařízení a obsahuje id, jméno, kategorii zařízení, seznam skupinových adres pro jednotlivé funkce zařízení a id místnosti do které zařízení patří.

`AddressType` představuje skupinovou adresu zařízení a skládá se z adresy a id příslušného `FunctionType`.

Pomocí třídy `FunctionType` se rozlišuje funkce skupinové adresy a skládá se z id, názvu, DPT a kategorie zařízení. Id tohoto objektu odpovídá enumeraci `FunctionTypeCategory`.



Obrázek 16: Databázový model

Kromě zmíněných tříd a ukládání a načítání dat z XML souboru Model obsahuje třídu **DataContext**. Ta poskytuje ViewModelům přístup k seznamu zařízení a místností. V případě, že si některý z ViewModelů vyžádá data poskytovaná **DataContextem**, dojde k jejich načtení z XML souboru a veškeré změny jsou pak prováděny pouze v **DataContextu**. Trvalé uložení proběhne až po vyžádání uživatelem.

### 6.3.2 ViewModel

ViewModel funguje jako pojítka mezi View a Modelem. Spolupracuje s Modelem tak, že volá jeho metody a získaná data poskytuje View ve tvaru, který View snadno zpracuje. Obsahuje také příkazy, které jsou vyvolány uživatelem ve View. Například pokud uživatel klikne ve View na tlačítko je zavolána metoda ve ViewModelu.

ViewModel informuje View o změnách pomocí události **PropertyChanged** z rozhraní **INotifyPropertyChanged**. Implementace rozhraní je popsána ve výpisu níže. V případě, že nastane změna některé property ViewModelu jejíž hodnota je zobrazována ve View, ViewModel o této změně informuje vyvoláním události **PropertyChanged** a data zobrazována ve View se automaticky obnoví.

---

```

public class INotifyBase : INotifyPropertyChanged
{
    public event PropertyChangedEventHandler PropertyChanged;
    public void RaisePropertyChanged([CallerMemberName] string propertyName =
        null)
    {
        if (PropertyChanged != null)
            PropertyChanged(this, new PropertyChangedEventArgs(propertyName));
    }
}

```

---

Výpis 1: Implementaci rozhraní INotifyPropertyChanged

V případě kolekcí se využívá `ObservableCollection`, tato třída automaticky informuje o změnách. V případě přidání nebo odebrání prvku z kolekce dojde k vyvolání události `CollectionChanged`. Prvky v kolekci musí implementovat rozhraní `INotifyPropertyChanged`.

Ve ViewModelu lze také reagovat na uživatelské akce prováděné ve View, jako je například stisknutí tlačítka. K tomu je využívání rozhraní `ICommand`. Pro zjednodušenou práci je toto rozhraní implementováno ve třídě `ICommandBase` a použití je následovné.

---

```

AddRoomCommand = new ICommandBase(AddRoom);
public void AddRoom(object parametr)
{
    DatabaseContext.Rooms.Add(new Room(true) { Name = NewRoom.Name });
}

```

---

Výpis 2: Použití rozhraní ICommand

### 6.3.3 View

View jsou jednotlivé obrazovky uživatelského rozhraní. Představují prezentační vrstvu aplikace a jsou tvořeny pomocí jazyka XAML. Veškerá logika se provádí ve ViewModelu, ten se v každém view definuje v elementu `Page.DataContext`.

---

```

<Page.DataContext>
    <ViemModel:ConfigurationViewModel/>
</Page.DataContext>

```

---

Výpis 3: Definování použitého ViewModelu

Pro navázání akce popsané výše na konkrétní tlačítko, je do elementu `Button` přidán atribut `Command="Binding AddRoomCommand"`.

Pro načtení kolekcí definovaných ve ViewModelu se využívá atribut `ItemsSource`. Za klíčovým slovem `Binding` je definována kolekce z ViewModelu, která se použije pro naplnění výpisu. V atributu `SelectedItem` se uvádí název property z ViewModelu, ve které bude představovat aktuálně vybraný prvek z kolekce. Nastavení atributu `Mode` na hodnotu `TwoWay` způsobí vyvolání změn jak v cílové tak ve zdrojové property.

---

```
<ListBox x:Name="listRooms" HorizontalAlignment="Left" Height="181" Margin="43,50,0,0" VerticalAlignment="Top" Width="233" ItemsSource="{Binding Rooms}" SelectedItem="{Binding SelectedRoom, Mode=TwoWay}" DisplayMemberPath="Name"/>
```

---

Výpis 4: Použití atributů `ItemsSource` a `SelectedItem`

Aplikace obsahuje celkem pět hlavních View. `MainView` je hlavní šablona aplikace, obsahuje navigační lištu s tlačítky pro přechod mezi samotnými obrazovkami aplikace, ty se zobrazují ve zbylé části `MainView`. `ControlView` slouží pro zobrazení místností a k nim příslušných zařízení a obsahuje ovládací prvky pro interakci s inteligentní budovou. `ConfigurationView` zobrazuje rozhraní pro zadání konfigurace místností a zařízení. `SettingsView` je obrazovka, která umožňuje individuální konfiguraci uživatelského prostředí aplikace. `HelpView` poskytuje rychlou nápovědu pro ovládání aplikace.

## IValueConvertor

Při tvorbě vzhledu aplikace může nastat situace, kdy je potřeba zobrazit některé části jen za určitých podmínek. V případě aplikace Hestia nastává tato situace v `ConfigurationView`, kde se v seznamu již nakonfigurovaných zařízení zobrazují pole podle typu zařízení. Pokud je zařízení typu osvětlení pak se zobrazí pouze pole pro editaci adres funkcí žaluzií. V případě, že je zobrazované zařízení typu žaluzie, zobrazí se pouze pole pro editaci adres pro ovládání žaluzií.

Toho lze docílit implementací rozhraní `IValueConvertor`. To obsahuje dvě metody a to metodu `Convert` a `ConvertBack`. Metoda `Convert` na základě vstupní hodnoty tj. typ zařízení vrátí odpovídající enumeraci `Visibility`. Metoda `ConvertBack` v tomto případě postrádá smysl a není ji nutné implementovat.

---

```
public class CategoryToVisibilityLightsConverter : IValueConverter
{
    public object Convert(object value, Type targetType, object parameter,
        string language){
        return ((int)value == (int)DeviceCategory.Lights) ? Visibility.Visible :
            Visibility.Collapsed;
    }
}
```

---

Výpis 5: Implementace `IValueConvertoru`

Třidu `CategoryToVisibilityLightsConverter` lze pak snadno použít přímo v `ConfigurationView`. Atribut `Visibility` elementu, který se má zobrazovat na základě typu zařízení, je nastaven s použitím `Converteru`. Tomu je předána informace o typu zařízení z elementu `cmbSelectedCategory`. `LightsVisibility` je statický zdroj představující třídu `CategoryToVisibilityLightsConverter`.

---

```
<Grid x:Name="gridLights" Grid.Column="2" Visibility="{Binding ElementName=
    cmbSelectedCategory, Path=SelectedIndex, Converter={StaticResource
    LightsVisibility}}">
```

---

Výpis 6: Použití třídy `CategoryToVisibilityLightsConverter`

## 6.4 Komunikace s KNX sběrnici

Aby bylo možné se sběrnici KNX komunikovat je nutné ke sběrnici připojit KNXnet/IP router. Ten umožňuje zasílání informací mezi zařízeními a sběrnici pomocí IP protokolu. V případě bezdrátové komunikace je potřeba KNXnet/IP router připojit k bezdrátovému routeru a zařízení (počítač, mobilní telefon), z něhož chceme komunikovat se sběrnici, musí být připojeno ke stejné síti jako KNXnet/IP router.

Je možné využít dva způsoby připojení ke sběrnici - KNX IP tunneling a KNX IP routing. KNX IP tunneling umožňuje komunikaci mezi jedním externím zařízením a KNX sběrnici (unicast). Toto je podobné připojení přes USB nebo sériové rozhraní. KNX IP routing využívá multicast adresu, na kterou jsou zasílány telegramy. KNX asociace má rezervovanou multicast adresu 224.0.23.12. Této adrese jednotlivá KNX zařízení naslouchají a případně vykonávají jim určené požadavky [35].

Pro komunikaci s KNX sběrnici lze pro .Net framework využít Falcon SDK, které je poskytováno přímo asociací KNX. Tuto knihovnu však nelze použít v případě vývoje UWP. Zvolila jsem tedy jiné řešení a to knihovnu `KNXLib.Portable`<sup>12</sup>, která je určena přímo pro použití při vývoji UWP.

Knihovna umožňuje připojení a odpojení od sběrnice a zasílání akcí na sběrnici. Při zapnutí aplikace dojde k připojení ke sběrnici, při zavírání aplikace je potřeba aplikaci od sběrnice naopak odpojit.

Pomocí metody `Action` jsou na sběrnici zasílány telegramy. Metodě je předán parametr se skupinovou adresou a parametr s daty, které se mají na sběrnici zaslat. Například pro rozsvícení světel se na skupinovou adresu pro ovládání osvětlení zašle booleanovská hodnota `true`.

---

```
GlobalContext.KNX.Action(lDevice.LightsOnOffAddress, true);
```

---

Výpis 7: Zapnutí osvětlení

Kromě datového typu `boolean` lze na sběrnici v parametru pro data zaslat také `byte` a nebo je možné využít v knihovně již předdefinované datové typy (DPT), které odpovídají oficiální

---

<sup>12</sup><https://github.com/Rimobul/KNXLib.Portable>



definici asociace KNX. DPT slouží k převodu hodnot ze vstupního intervalu na data, která dokáže sběrnice zpracovat.

## 6.5 Hlasové ovládání

Pro rozpoznání řeči je využito Microsoft Speech Platform SDK 11. Rozpoznání řeči lze realizovat v několika jazycích. V této práci se využívá angličtina.

Rozpoznání hlasu funguje tak, že Speech Recognition Engine převede vstupní zvukovou stopu na text. Zvukový vstup je rozdělen do segmentů, které by mohly být řečí a dojde k převodu do série číselných hodnot reprezentujících zvukový signál. Takto upravená vstupní data jsou dále vyhodnocována oproti třem databázím<sup>13</sup>: akustickému modelu, lexikonu a jazykovému modelu. Akustický model představuje akustický projev jazyka a může se přizpůsobit rozpoznávání specifických rysů řeči jednotlivých uživatelů. Lexikon obsahuje velké množství slov v daném jazyce a poskytuje informace o jejich výslovnosti. Jazykový model poskytuje informaci o tom jakým způsobem lze slova kombinovat.

Zatímco zabudovaný jazykový model má reprezentovat rozsáhlou jazykovou doménu (jako například každodenní angličtina), aplikace většinou bude zpracovávat jen určité fráze, které mají konkrétní význam pro daný kontext. Než by aplikace měla využívat celý rozsah nabízeného jazyka, je lepší když používá pouze omezenou gramatiku vzhledem k účelu aplikace. Definování gramatiky má několik výhod: zvyšuje přesnost rozpoznání, zajišťuje že všechny výsledky jsou pro aplikaci smysluplné, umožňuje Speech Recognition Engine specifikovat sémantické hodnoty vzešlé z rozpoznávaného textu. Tuto gramatiku lze definovat programaticky nebo ve formě XML struktury splňující SRGS specifikaci.

### 6.5.1 Inicializace rozhraní pro rozpoznání hlasu

Při spuštění aplikace proběhne inicializace hlasového rozhraní pomocí funkce `OnLoadMainViewModel`, kde se volají funkce `InitializeRecognizer` a poté `RecognitionStart`. Dojde k vytvoření instance `SpeechRecognizer`, té je přidělena gramatika, na kterou má reagovat a je spuštěn proces naslouchání zvukovému vstupu.

---

```
private async void OnLoadMainVieModel()
{
    bool permissionGained = await AudioCapturePermissions.
        RequestMicrophonePermission();
    if (permissionGained){
        await SpeechContext.InitializeRecognizer();
        SpeechContext.RecognitionStart();
    }
}
```

---

<sup>13</sup>[https://msdn.microsoft.com/en-us/library/hh378337\(v=office.14\).aspx](https://msdn.microsoft.com/en-us/library/hh378337(v=office.14).aspx)

---

#### Výpis 8: Inicializace rozhraní pro hlasové ovládání

Po spuštění aplikace proběhne kontrola, jestli je k dispozici zařízení pro zaznamenání zvuku a zda má aplikace přístup k tomuto zařízení. Pokud tomu tak není, pak je uživateli zobrazeno upozornění. Jedná se pouze o informativní upozornění, aplikace je funkční i bez spuštěného rozhraní pro hlasové ovládání.

Dále se provádí inicializace Speech Recognition Engine v metodě `InitializeRecognizer()`, v které dojde k vytvoření instance třídy `SpeechRecognizer`. Té je předán parametr s informací o aktuálním jazyce pro řeč v systému. V české verzi Windows 10 je hodnota parametru `en-US`, je tedy nastavena angličtina (spojené státy).

---

```
SpeechRecognizer = new SpeechRecognizer(SpeechRecognizer.SystemSpeechLanguage);
StorageFile grammarContentFile = await Package.Current.InstalledLocation.
    GetFileAsync("SRGS\\SRGS.xml");
SpeechRecognitionGrammarFileConstraint grammarConstraint = new
    SpeechRecognitionGrammarFileConstraint(grammarContentFile);
SpeechRecognitionCompilationResult result = await SpeechRecognizer.
    CompileConstraintsAsync();
```

---

#### Výpis 9: Průběh inicializace

Načte se soubor obsahující gramatiku pro hlasové rozpoznávání. Ten je předán instanci `SpeechRecognizer` a provede se kompilace gramatiky ve formátu XML. Pokud by gramatika v XML souboru nebyla ve správném tvaru, Speech Recognition Engine gramatiku nepřijme a inicializace rozhraní pro hlasové ovládání nebude úspěšná.

Pokud dojde ke spuštění rozhraní pro hlasové ovládání, pak se ve stavovém řádku zobrazí zelená ikona mikrofону s informací, že rozhraní je připraveno. V opačném případě bude ikona červená a zobrazí se informace, že hlasové ovládání není k dispozici.

V případě, že hlasové ovládání je spuštěno, je možné jej pomocí kliknutí na mikrofón vypnout a aplikace přestane naslouchat.

### 6.5.2 Speech Recognition Grammar Specification

Jedná se o syntaxi reprezentující gramatiku používanou pro rozpoznání hlasu. Tuto syntaxi je možné vytvořit ve dvou formátech a to buď ve formátu Augmented BNF nebo v XML formátu, které jsou vzájemně převoditelné.

Gramatika se skládá z pravidel, které obsahují slova a fráze rozpoznatelné aplikací. V případě gramatiky ve formátu XML je kořenovým elementem element `grammar`.

---

```
<grammar mode="voice" root="main" version="1.0" tag-format="semantics/1.0" xml:
  lang="en-US" xmlns="http://www.w3.org/2001/06/grammar">

</grammar>
```

---

#### Výpis 10: Element grammar

Veškerý obsah gramatiky musí být umístěn mezi počátečním a koncovým tagem elementu **grammar**. Kořenový element obsahuje následující povinné a nepovinné atributy.

- **mode** - Jedná se o nepovinný atribut, který může nabývat hodnot **voice**- pro mluvené slovo a **dtmf** - pro tónovou volbu. Pokud je vynechán, výchozí hodnota je **voice**.
- **root** - Volitelný, ale doporučený atribut, který určuje, jaké pravidlo bude aktivováno při načtení gramatiky Speech Recognition Enginem. Pokud není označeno žádné pravidlo jako kořenové, dojde k validaci a kompilaci gramatiky, ale rozpoznání nebude spuštěno.
- **tag-format** - Povinný atribut v případě, že gramatika obsahuje elementy s názvem **tag**. Specifikuje jakého typu je obsah všech elementů **tag** vyskytujících se v gramatice. Může nabývat třech možných hodnot - **semantics/1.0** - obsahem je ECMAScript, **semantics-ms/1.0** - obsahem je ECMAScript implemantovaný Microsoftem, **semantics/1.0-literals** - obsahem je boolovská hodnota, celé číslo, číslo s plovoucí desetinnou čárkou nebo řetězec, který se nesmí nacházet v dvojitéch uvozovkách.
- **version**- Povinný atribut, který specifikuje označení použité verze SRGS. Jediná přijatelná hodnota je **1.0**.
- **xml:lang** - Povinný atribut v případě, že je zvolen mode voice. Definuje jazyk použit pro vytvoření gramatiky.
- **xmlns**- Povinný atribut, který definuje XML jmenný prostor pro gramatiku a jeho hodnota je **http://www.w3.org/2001/06/grammar**.

Kořenový element **grammar** může obsahovat další atributy, které nejsou v této konkrétní gramatice využity. Jedná se o atributy **sapi:alphabet**, **xml:base**, **xmlns:sapi** [26].

Každá gramatika musí obsahovat alespoň jedno pravidlo. To se vkládá do elementu **rule**. Pravidlo obsahuje slovo nebo frázi, kterou může uživatel říci. Každé pravidlo má povinný atribut **id**, ten se využívá pro referencování pravidla v ostatních pravidlech.

Slovo nebo fráze je umístěno uvnitř elementu **item**. V případě že potřebujeme, aby bylo z pravidla přijata vždy pouze jedna možnost, vložíme jednotlivé položky mezi počáteční a koncový element **one-of**.

---

```
<rule id="Number">
  <one-of>
    <item>
      zero
    </item>
    <item>
      one
    </item>
    <item>
      two
    </item>
  </one-of>
</rule>
```

---

Výpis 11: Ukázka jednoduchého pravidla

Pravidla je možné referencovat z ostatních pravidel a to pomocí elementu **ruleref** a atributu **uri**. Lze tak tvořit složitější strukturu SRGS.

---

```
<ruleref uri="#Number"/>
```

---

Výpis 12: Odkaz na pravidlo

Kromě toho, že lze pomocí gramatiky poznat, zda uživatel vyslovil některý z definovaných příkazů a získat jej v textové podobě. Lze také získat sémantické hodnoty k jednotlivým příkazům pomocí elementu **tag** a klíčového slova **out**.

---

```
<rule id="Number" scope="public">
  <one-of>
    <item>
      zero <tag>out = 0; </tag>
    </item>
    <item>
      one <tag>out = 1; </tag>
    </item>
    <item>
      two <tag>out = 2; </tag>
    </item>
  </one-of>
</rule>
```

---

Výpis 13: Pravidlo se sémantickými hodnotami

Pomocí volitelného atributu `repeat` elementu `item`, je možné určit počet opakování slova. V případě zadání 0-1 se slovo ve vyslovené větě nemusí vyskytovat a přesto bude pravidlo rozpoznáno.

---

```
<item repeat="0-1">
  <ruleref uri="#Number"/>
  <tag>out.Number = rules.Number</tag>
</item>
```

---

Výpis 14: Ukázka opakování výrazu

V případě, že uživatel vysloví větu definovanou v gramatice, bude vyvolána událost `ResultGenerated`. Ta dále zpracovává výsledek rozpoznáního povelu. Jednotlivé sémantické hodnoty lze získat zavoláním

```
recoResult.SemanticInterpretation.Properties["Number"][0].ToString()
```

---

```
<rule id="PossibleCombination">
  <one-of>
    <item>
      <ruleref uri="#Number"/>
      <tag>out = rules.Number</tag>
    </item>
    <item>
      <ruleref uri="#Number"/>
      <tag>out.One = rules.Number</tag>
      <ruleref uri="#Number"/>
      <tag>out.Two = rules.Number</tag>
      <tag>out = out.One + " " + out.Two</tag>
    </item>
    <item>
      <ruleref uri="#Number"/>
      <tag>out.One = rules.Number</tag>
      <ruleref uri="#Number"/>
      <tag>out.Two = rules.Number</tag>
      <ruleref uri="#Number"/>
      <tag>out.Three = rules.Number</tag>
      <tag>out = out.One + " " + out.Two + " " + out.Three</tag>
    </item>
  </one-of>
```

---

Výpis 15: Definování možností zadání číselného vstupu

S využitím výše zmíněných pravidel lze například z nadiktovaného vstupu „*two five six slash two seven slash six*“, získat tříúrovňovou skupinovou adresu zařízení ve tvaru 256/27/6. Pro zjednodušení zadávání čísel je použito diktování jednotlivých číslic za sebou.

V gramatice je definováno, že uživatel musí vyslovit tři skupiny čísel oddělené lomítkem (výraz „slash“), každá skupina čísel může obsahovat 1-3 číslice. V případě hlasového zadávání není zohledněn tvar tříúrovňové skupinové adresy 0-255/0-255/0-255. Tedy pokud uživatel vysloví správnou posloupnost čísel a lomítek takovou, že bude vygenerováno např. číslo 999/999/999, bude Speech Recognition Enginem rozpoznáno, ale při pokusu o nastavení adresy ve ViewModelu dojde k validaci adresy a uživatel bude upozorněn, že adresa byla zadána ve špatném tvaru.

V rámci každé skupiny čísel je možné říci jednu, dvě nebo tři číslice. Toto je třeba v gramatice zohlednit, lze to udělat podobně jako ve výpisu 15.

Pravidlo definuje, že v případě zadání pouze jedné číslovky, je vrácena sémantická hodnota přímo této číslovky z pravidla `Number`, které vrací hodnotu vysloveného čísla z množiny 0 až 9. V případě zadání dvou čísel je vráceno jejich zřetězení, taktéž při zadání tří čísel. Skupinu čísel nelze zadat jinak než jednou ze zmíněných variant.

Definování celé skupinové adresy je realizováno následujícím pravidlem:

---

```
<rule id="AddressFormat" scope="public">
  <item>
    <ruleref uri="#PossibleCombination"/>
    <tag>out.First = rules.PossibleCombination</tag>
  </item>
  <item>slash</item>
  <item>
    <ruleref uri="#PossibleCombination"/>
    <tag>out.Second = rules.PossibleCombination</tag>
  </item>
  <item>
    <item>slash</item>
    <ruleref uri="#PossibleCombination"/>
    <tag>out.Third = rules.PossibleCombination</tag>
  </item>
  <tag>out = out.First + "/" + out.Second + "/" + out.Third</tag>
</rule>
```

---

Výpis 16: Pravidlo pro zadání skupinové adresy

Aby byla posloupnost vyslovených číslic přijata Speech Recognition Enginem, musí být nejprve vyslovena první skupina čísel dle definice v `PossibleCombination` následována povinným slovem „slash“. Dále druhá skupina čísel v jednom z možných tvarů, opět slovo „slash“ a poslední skupina čísel.

Takto je možné získat skupinovou adresu ve tvaru 0-999/0-999/0-999 přímo z vyslovené věty. Na pravidlo `AddressFormat` je odkazováno v nadřazeném pravidle a je zde definován název property, do které bude uložena sémantická informace.

---

```
<ruleref uri="#AddressFormat"/>
<tag>out.Address = rules.AddressFormat;</tag>
```

---

Výpis 17: Odkaz na pravidlo `AddressFormat` a jeho sémantický výstup

### 6.5.3 Konkrétní příklad zpracování hlasového povelu

V sekvenčním diagramu na obrázku 17 je popsána situace, kdy uživatel pomocí hlasu zadává skupinovou adresu v konfiguračním nástroji. Konkrétní situace může vypadat následovně:

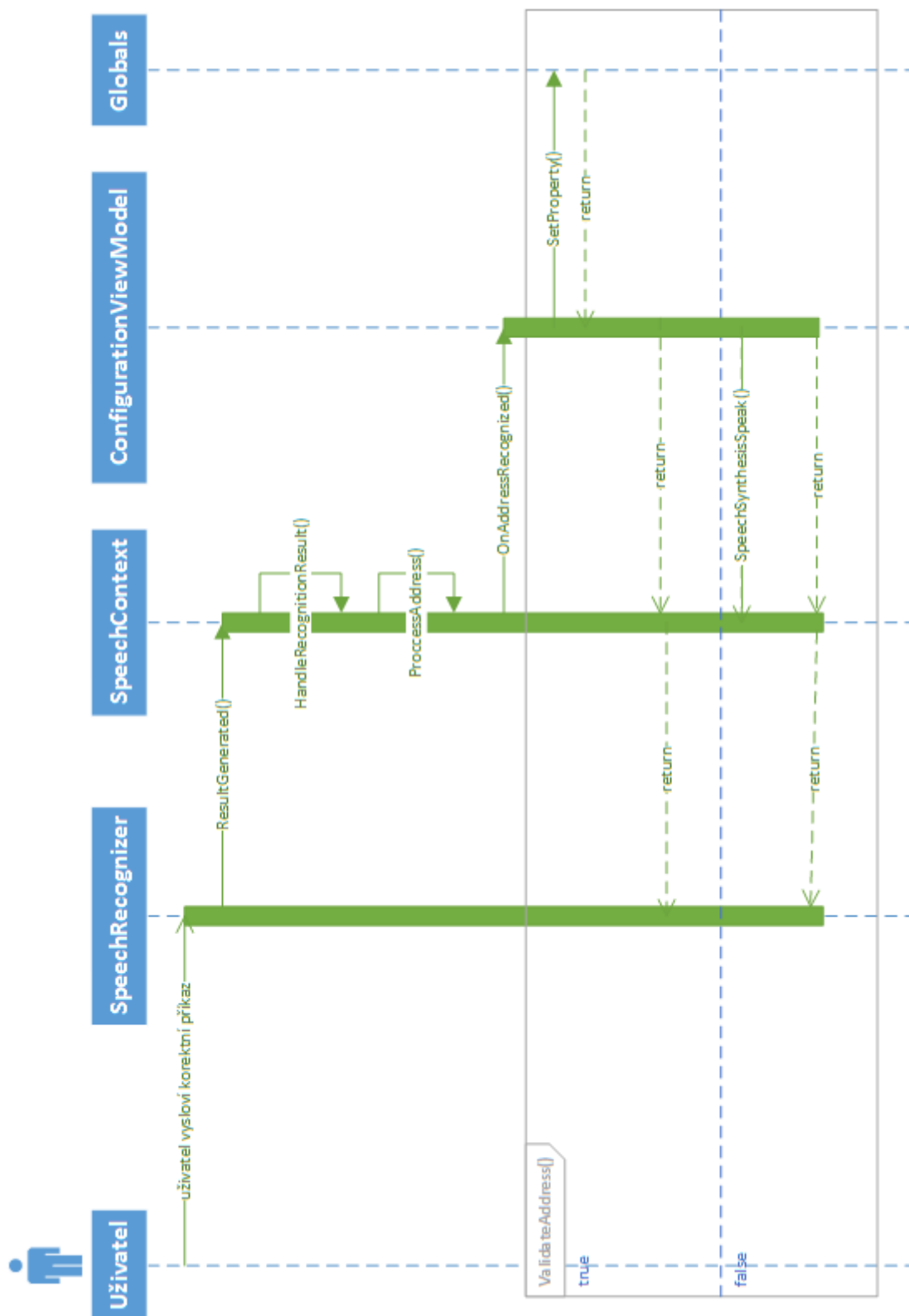
Uživatel vysloví příkaz „*write lights on off address one slash two slash three*“. Pokud příkaz vysloví s dostatečně dobrou výslovností, Speech Recognition Engine rozpozná, že se jedná o příkaz z gramatiky a vyvolá se událost `ResultGenerated`. Výsledek, který je vytvořen obsahuje informaci s jakou jistotou byl příkaz rozpoznán. K tomu slouží enumerace

`SpeechRecognitionConfidence`, která může nabývat čtyř stavů - `High`, `Medium`, `Low` a `Rejected`. V případě, že příkaz byl rozpoznán s vysokou nebo střední jistotou, je dále zavolána metoda `HandleRecognitionResult`, která zpracovává sémantické hodnoty výsledku. V metodě `HandleRecognitionResult` dochází k prvotnímu zpracování výsledku. Na základě hodnoty sémantické property `Type` se zavolá příslušná metoda pro další zpracování výsledku. V tomto případě bude tato hodnota `address` a další zpracování výsledku bude pokračovat v metodě `ProcessAddress`.

Metoda `ProcessAddress` slouží k načtení všech sémantických hodnot z vysloveného příkazu. Jedná se o adresu samotnou, její typ a informaci jestli se bude upravovat nové nebo existující zařízení. Pokud jsou k dispozici všechna tato data, vyvolá se událost `OnAddressRecognized`.

Na událost `OnAddressRecognized` je v `ConfigurationViewModelu` navázána metoda `SpeechContext_OnAddressRecognized`. Nejprve se provede validace adresy. V případě, že adresa není validní, dojde k zavolání metody `SpeechSynthesisSpeak` s parametrem chybového hlášení a pomocí syntézy hlasu je uživatel upozorněn, že adresu zadal ve špatném formátu. V případě validní adresy je zavolána metoda `SetProperty`, která pomocí reflexe nastaví konkrétní skupinovou adresu. Po úspěšném nastavení adresy je vyvolána událost `OnPropertyChanged`, která informuje View o změnách ve ViewModelu a View tyto změny zobrazí.

Reflexe se také využívá pro realizaci ovládání pomocí hlasu. V gramatice jsou definovány názvy metod, které by se měli vyvolat na základě vysloveného povelu. Pokud tedy uživatel např. řekne příkaz „*light on*“, z gramatiky je vrácena informace, že se má volat metoda `LightOn`, která slouží pro zapínání osvětlení. Jedná se o stejnou metodu, která se vyvolá po stisknutí tlačítka „Zapnout“.



Obrázek 17: Sekvenční diagram popisující zpracování hlasového příkazu



## 6.6 Přizpůsobení pro mobilní telefony

Univerzální aplikace pro Windows 10 sdílí společný kód. Je však potřeba přizpůsobit uživatelské rozhraní pro zobrazení na různých zařízeních. Toho lze dosáhnout použitím třídy `VisualStateManager`.



Obrázek 18: Aplikace v mobilním zařízení

Obrazovka sloužící k ovládání budovy je tvořena levou částí se seznamem místností a pravou částí s detailem jednotlivých místností obsahujících seznam zařízení. Pokud se šířka okna, ve kterém je spuštěna aplikace, zmenší pod 720 pixelů, zobrazí se pouze seznam místností a po kliknutí na konkrétní místnost, je uživatel přesměrován na novou obrazovku s detailem místností (viz. obr. 18).

Tato funkcionality je realizován tak, že v `ControlView.xaml` je definována element `VisualStateManager`. V něm jsou vytvořeny dva stavy pomocí elementu `VisualState` - stav `Default` a `Reduced`. Pomocí elementu `AdaptiveTrigger` lze definovat podmínky použití jednotlivých stavů a pomocí elementu `Setter` je možné měnit hodnoty již existujících elementů a jejich atributů v XAML souboru.

`VisualStateManager` dokáže reagovat jak na změnu šířky, tak na změnu výšky. Je tedy možné v elementu `AdaptiveTrigger` použít buď atribut `MinWindowWidth`, nebo `MinWindowHeight`. Při šířce okna větší než 720 pixelů se bude vzhled obrazovky měnit podle nastavených atributů ve `VisualState` elementu s názvem `Default`. Při šířce okna od 0 - 720 pixelů naopak z elementu `VisualState` s názvem `Reduced`.

Při zobrazení aplikace v zařízení s šířkou nižší než 720 pixelů, nebo při zmenšení okna aplikace, je nastavena šířka levé části obrazovky, ve které se nachází výpis místností, na plnou šířku okna. Naopak šířka pravé části, kde se nachází detail místnosti se seznamem zařízení, je nastavena na nulovou hodnotu, nebude se tedy zobrazovat vůbec.

---

```
<VisualStateManager.VisualStateGroups>
  <VisualStateGroup x:Name="States" CurrentStateChanged="
    States_CurrentStateChanged">
    <VisualState x:Name="Default">
      <VisualState.StateTriggers>
        <AdaptiveTrigger MinWindowWidth="720" />
      </VisualState.StateTriggers>
    </VisualState>
    <VisualState x:Name="Reduced">
      <VisualState.StateTriggers>
        <AdaptiveTrigger MinWindowWidth="0"/>
      </VisualState.StateTriggers>
      <VisualState.Setters>
        <Setter Target="Master.Width" Value="*" />
        <Setter Target="Detail.Width" Value="0" />
        <Setter Target="RoomListView.SelectionMode" Value="None" />
      </VisualState.Setters>
    </VisualState>
  </VisualStateGroup>
</VisualStateManager.VisualStateGroups>
```

---

Výpis 18: Definování `VisualStateManager`u

Pokud by při úplném zobrazení, tedy nad 720 pixelů, byla vybrána některé z místností, pak při zmenšení okna nebude zobrazen výpis místnosti, ale již detail označené místnosti. V tomto případě je potřeba změnit aktuálně zobrazené `ControlView` přímo na `ControlDetailView`. Toto je nutné explicitně naprogramovat a lze toho dosáhnout z codebehind při vyvolání události `CurrentStateChanged`. Ta se vyvolá při splnění podmínek některého z adaptivních triggerů.

## 7 Testování

Testování aplikace probíhalo na výukovém panelu Schneider, který se skládá ze čtyř dílčích panelů. První panel obsahuje výměnné šablony s půdorysem a diody simulující osvětlení a žaluzie. Druhý panel obsahuje tlačítkové snímače, pokojové termostaty a snímače pohybu. Třetí panel obsahuje reálnou žaluzii, snímač venkovního jasu a teploty. Čtvrtý panel obsahuje 10" dotykovou obrazovku.

Pro účely testování aplikace je podstatný panel s reálnou žaluzií a panel s půdorysem a diodami simulujícími osvětlení. Program nahráný na panelu odpovídá diplomové práci [36].

### 7.1 Ovládání provozně technických funkcí

Při testování ovládání provozně technických funkcí na výukovém panelu byli pro ovládání osvětlení testovány funkce zapínání a vypínání osvětlení a stmívání. Pro ovládání žaluzií bylo testováno ovládání pohybu žaluzií nahoru a dolů, pohyb do určité výšky žaluzie a změna náklonu lamel žaluzie.

V případě testování zapínání a vypínání světel byli na skupinovou adresu aktoru (v ETS označeno jako switch object) zaslány booleanovské hodnoty true a false. Zaslání těchto hodnot je vyvoláno stisknutím tlačítka zapnout resp. vypnout v obrazovce určené pro ovládání. Tyto tlačítka jsou k dispozici při zobrazení místnosti, ve které se nachází zařízení s vyplněnou skupinovou adresou pro tuto funkci. Při stisknutí tlačítka zapnout byl na sběrnici zaslán telegram s cílovou skupinovou adresou a hodnotou true. Aktor s touto skupinovou adresou odposlechl telegram a rozsvítil diodu na panelu. Stejným způsobem bylo otestováno vypínání světel, kdy byla zaslána na sběrnici hodnota false a došlo k zhasnutí světla.

Pro funkci stmívání (v ETS označeno jako dimming object) jsou v ovládacích prvcích k dispozici tlačítka s označením minus (-) a plus (+). Po stisknutí tlačítka plus dojde k zaslání kladné předdefinované hodnoty na sběrnici a dojde ke zvýšení jasu diody. Při stisknutí tlačítka minus se zašle na skupinovou adresu stmívacího aktoru záporná předdefinovaná hodnota a dojde ke snížení jasu diody.

V případě ověřování funkce pohybu žaluzií nahoru a dolů (v ETS označeno jako movement object for manual mode) byli v ovládací části aplikaci stisknuty tlačítka pro pohyb nahoru resp. dolů. Došlo ke stejným akcím na sběrnici jako v případě zapínání a vypínání světel.

Při realizaci pohybu žaluzií do určité výšky (v ETS označeno jako height position in manual mode), nebo při nastavení náklonu lamel (v ETS označeno jako slat in manual mode), se v aplikaci zadává hodnota z rozmezí 0-100%. Tato hodnota je převedena na číslo z intervalu 0-255 a tato data jsou následně zaslána na nakonfigurovanou adresu pro danou funkci.

Veškeré funkce pro ovládání, které aplikace poskytuje, byly otestovány se stoprocentní spolehlivostí.

## 7.2 Hlasové ovládání

Testování hlasového ovládání bylo provedeno s pomocí skupiny 8 osob ve věku 25-73 let různého pohlaví. Skupina postupně vyslovovala několik povelů podstatných pro ovládání. Každý z povelů byl třikrát zopakován. Výsledky testování jsou zaznamenány v grafu na obrázku 19.

Pro správné rozpoznání povelu byla důležitá vzdálenost od mikrofону, dále ticho v místnosti a správná výslovnost anglických výrazů.



Obrázek 19: Úspěšnost rozpoznání hlasových příkazů

## 8 Závěr

Tato práce se zabývala vývojem aplikace pro ovládání provozně technických funkcí v inteligentních budovách vybavených sběrníkovým systémem KNX. Výsledkem je univerzální aplikace pro Windows 10, kterou je možné spustit jak na desktopové verzi systému, tak na mobilní verzi operačního systému Windows 10.

V práci je popsáno využití architektonického vzoru MVVM pro vývoj UWP. V aplikaci bylo implementováno hlasové ovládání, které je k dispozici v anglickém jazyce a slouží k usnadnění ovládání aplikace nejen starším lidem a handicapovaným. Aplikace obsahuje několik nastavení ulehčující práci s aplikací jak seniorům, tak i ostatním uživatelům.

Zadání obsahuje kromě praktické části také zpracování teoretických bodů, týkajících se porovnání sběrníkových systému a vizualizačních nástrojů ve Smart Home Care. Tyto teoretické části byli v práci zpracovány.

Funkčnost aplikace byla ověřena na výukovém panelu Schneider, kde bylo otestováno ovládání osvětlení a žaluzií. Ověřena byla také funkčnost hlasového ovládání.

Oproti jiným pracím zabývajících se tvorbou prostředí pro ovládání inteligentní budovy, je výsledná aplikace konfigurovatelná a je tedy použitelná pro jakoukoli budovu vybavenou sběrníkovým systémem KNX. Implementace hlasového ovládání je vlastním řešením, není zde využito softwaru třetích stran.

Aplikaci by bylo možno dále rozšířit o zobrazení stavů ovládaných zařízení. Také by bylo možné implementovat komplexní příkazy (např. zastření žaluzií a následné rozsvícení světel v místnosti), nebo naplánované úlohy (např. zastření všech žaluzií po 17h). Dalším rozšířením by mohlo být propojení s cloudovým uložištěm pro uchování aktuální konfigurace a implementace přihlášení do aplikace. Zjednodušila by se konfigurace aplikace, která by proběhla pouze jednou a bylo by možné aplikaci spustit se stejnou konfigurací z více zařízení.

## Literatura

- [1] HARPER, Richard (ed.). *Inside the smart home*. London: Springer, c2003. ISBN 18-523-3688-9.
- [2] *If You Can't Stand the Coding, Stay Out of the Kitchen: Three Chapters in the History of Home Automation* [online]. [cit. 2015-08-05]. Dostupné z: <http://www.drdoobbs.com/architecture-and-design/if-you-cant-stand-the-coding-stay-out-of/184404040>
- [3] BOUNEGRU, Liliana. *Smart Houses: From Managing the House at a Distance to the Management of Life Itself* [online]. [cit. 2015-08-05]. Dostupné z: [http://lilianabounegrue.org/wp-content/uploads/2009/11/thesis\\_smart\\_houses\\_liliana\\_bounegrue.pdf](http://lilianabounegrue.org/wp-content/uploads/2009/11/thesis_smart_houses_liliana_bounegrue.pdf)
- [4] DRISCOLL, Edward. *The history of X10* [online]. [cit. 2015-08-05]. Dostupné z: [http://home.planet.nl/~lhendrix/x10\\_history\\_english.htm](http://home.planet.nl/~lhendrix/x10_history_english.htm)
- [5] CHRISTENSEN, Kaare, Gabriele DOBLHAMMER, Roland RAU a James W VAUPEL. Ageing populations: the challenges ahead. *The Lancet*. 2009, 374(9696), 1196-1208. DOI: 10.1016/S0140-6736(09)61460-4. ISSN 01406736. Dostupné také z: <http://linkinghub.elsevier.com/retrieve/pii/S0140673609614604>
- [6] MCCREADIE, CLAUDINE a ANTHEA TINKER. The acceptability of assistive technology to older people. *Ageing and Society*. 1999, 25(1), 91-110. DOI: 10.1017/S0144686X0400248X. ISSN 0144-686x. Dostupné také z: [http://www.journals.cambridge.org/abstract\\_S0144686X0400248X](http://www.journals.cambridge.org/abstract_S0144686X0400248X)
- [7] MALINOVÁ, Nikol. *Vizualizace provozně technických funkcí v inteligentních budovách s asistivní péčí*. 2015. Diplomová práce. VŠB – Technická univerzita Ostrava
- [8] MISKELLY, F. G. Assistive technology in elderly care. *Age and Ageing*. 30(6), 455-458. DOI: 10.1093/ageing/30.6.455. ISSN 14682834. Dostupné také z: <http://www.ageing.oupjournals.org/cgi/doi/10.1093/ageing/30.6.455>
- [9] SPOSARO, F. a G. TYSON. IFall: An android application for fall monitoring and response. *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2009, , 6119-6122. DOI: 10.1109/IEMBS.2009.5334912. Dostupné také z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5334912>
- [10] BLASCO, Rubén, Álvaro MARCO, Roberto CASAS, Diego CIRUJANO a Richard PICKING. A Smart Kitchen for Ambient Assisted Living. *Sensors*. 2014, 14(1), 1629-1653. DOI: 10.3390/s140101629. ISSN 1424-8220. Dostupné také z: <http://www.mdpi.com/1424-8220/14/1/1629/>

- [11] CHAN, Marie, Daniel ESTÈVE, Christophe ESCRIBA a Eric CAMPO. A review of smart homes—Present state and future challenges. *Computer Methods and Programs in Biomedicine*. 2008, 91(1), 55-81. DOI: 10.1016/j.cmpb.2008.02.001. ISSN 01692607. Dostupné také z: <http://linkinghub.elsevier.com/retrieve/pii/S0169260708000436>
- [12] Technické informace o KNX / EIB systému. *Řízení stínící techniky, pohony bran a vrat - SOMFY Česká republika* [online]. [cit. 2016-03-29]. Dostupné z: [http://www.somfyarchitecture.cz/downloads/buildings/technicke\\_informace\\_o\\_knx\\_systemu.pdf](http://www.somfyarchitecture.cz/downloads/buildings/technicke_informace_o_knx_systemu.pdf)
- [13] MERZ, Hermann, Thomas HANSEMANN a Christof HÜBNER. *Automatizované systémy budov: sdělovací systémy KNX/EIB, LON a BACnet*. 1. vyd. Praha: Grada, 2008, 261 s. ISBN 978-80-247-2367-9.
- [14] ETS4 eCampus [online]. [cit. 2016-03-29]. Dostupné z: <http://wbt4.knx.org/>
- [15] VOJÁČEK, Antonín. *Sběrnice LonWorks - 1.část - Úvod*. In: [online]. [cit. 2016-03-29]. Dostupné z: <http://automatizace.hw.cz/clanek/2005040501>
- [16] MATZ, Václav. *Typy sběrnic a protokolů používaných pro komunikaci systémů* [online]. [cit. 2016-03-29]. Dostupné z: [http://www.ib.cvut.cz/sites/default/files/Honeywell\\_prednasky/Protokoly\\_1.pdf](http://www.ib.cvut.cz/sites/default/files/Honeywell_prednasky/Protokoly_1.pdf)
- [17] VAŇUŠ, Jan. *Popis sběrniceového systému Nikobus*. [online]. [cit. 2016-03-29]. Dostupné z: <http://feil.vsb.cz/kat420/vyuka/Bakalarske/STB/3.%20popis%20sber.%20syst.%20%20Nikobus%20promitani.pdf>
- [18] *Inels a sběrnice CIB – moderní systém inteligentní elektroinstalace*. [online]. [cit. 2016-03-29]. Dostupné z: <http://automa.cz/res/pdf/38218.pdf>
- [19] *Možnosti ovládání. IBSOLUTION* [online]. [cit. 2016-03-29]. Dostupné z: <http://www.ibsolution.cz/intelligentni-dum/moznosti-ovladani/>
- [20] *INels Home Control: První aplikace pro ovládání domácnosti ze Smart TV*. [online]. [cit. 2016-03-29]. Dostupné z: [http://www.stech.cz/Portals/0/Konference/2014/09-19%20IDD/prezentace/02\\_tobolik.pdf](http://www.stech.cz/Portals/0/Konference/2014/09-19%20IDD/prezentace/02_tobolik.pdf)
- [21] PRŮCHA, Jan. *Chytré bydlení: Inteligentní dům*. In: InsightHome [online]. [cit. 2016-03-29]. Dostupné z: <http://www.insighthome.eu/Chytre-bydleni/Chytre-bydleni.pdf>
- [22] *OpenRemote* [online]. [cit. 2016-03-29]. Dostupné z: <http://www.openremote.org/display/HOME/OpenRemote>

- [23] FRANCO, C., A. FLEURY, P. Y. GUMERY, B. DIOT, J. DEMONGEOT a N. VUILLERME. IBalance-ABF: A Smartphone-Based Audio-Biofeedback Balance System. *IEEE Transactions on Biomedical Engineering*. 2013, 60(1), 211-215. DOI: 10.1109/TBME.2012.2222640. ISSN 0018-9294. Dostupné také z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6320617>
- [24] CHERNBUMROONG, Saisakul, Shuang CANG, Anthony ATKINS a Hongnian YU. Elderly activities recognition and classification for applications in assisted living. *Expert Systems with Applications*. 2013, 40(5), 1662-1674. DOI: 10.1016/j.eswa.2012.09.004. ISSN 09574174. Dostupné také z: <http://linkinghub.elsevier.com/retrieve/pii/S0957417412010585>
- [25] *KNX Communication* [online]. [cit. 2016-03-29]. Dostupné z: [http://www.knx.org/fileadmin/template/documents/downloads\\_support\\_menu/KNX\\_tutor\\_seminar\\_page/basic\\_documentation/Communication\\_E1212a.pdf](http://www.knx.org/fileadmin/template/documents/downloads_support_menu/KNX_tutor_seminar_page/basic_documentation/Communication_E1212a.pdf)
- [26] Create an XML Grammar Structure (Microsoft.Speech). *MSDN* [online]. [cit. 2016-03-29]. Dostupné z: [https://msdn.microsoft.com/en-us/library/hh378465\(v=office.14\).aspx](https://msdn.microsoft.com/en-us/library/hh378465(v=office.14).aspx)
- [27] Guide to Universal Windows Platform (UWP) apps. *MSDN* [online]. [cit. 2016-03-29]. Dostupné z: <https://msdn.microsoft.com/en-us/library/windows/apps/dn894631.aspx#introduction>
- [28] Separate UI and app logic using the Model-View-ViewModel pattern. *MSDN* [online]. [cit. 2016-03-29]. Dostupné z: [https://msdn.microsoft.com/en-us/library/windows/apps/jj721615\(v=vs.105\).aspx](https://msdn.microsoft.com/en-us/library/windows/apps/jj721615(v=vs.105).aspx)
- [29] Serial Data Transmission and KNX Protocol. *KNX Association* [online]. [cit. 2016-03-29]. Dostupné z: [http://www.knx.org/fileadmin/template/documents/downloads\\_support\\_menu/KNX\\_tutor\\_seminar\\_page/tutor\\_documentation/05\\_Serial%20Data%20Transmission\\_E0808f.pdf](http://www.knx.org/fileadmin/template/documents/downloads_support_menu/KNX_tutor_seminar_page/tutor_documentation/05_Serial%20Data%20Transmission_E0808f.pdf)
- [30] Control unit with touch-screen display EST-2/W. *ELKO EP* [online]. [cit. 2016-03-30]. Dostupné z: <http://www.elkoep.com/products/inels-bus-system/inels-2/wall-controllers/control-unit-with-touch-screen-display-est-2w-12730/>
- [31] *DIVUS* [online]. [cit. 2016-03-30]. Dostupné z: <http://www.divus.eu/>
- [32] *LOXONE* [online]. [cit. 2016-03-30]. Dostupné z: <http://www.loxone.com/cscz/start.html>
- [33] KNX Dashboard. *Microsoft Store* [online]. [cit. 2016-03-30]. Dostupné z: <https://www.microsoft.com/cs-cz/store/apps/knx-dashboard/9wzdnrcdm06f>
- [34] *HomeGenie* [online]. [cit. 2016-03-30]. Dostupné z: <http://www.homegenie.it/index.html>



- [35] Solutions: IP and KNX – Bringing You up to Speed. *KNX today* [online]. 2014 [cit. 2016-03-30]. Dostupné z: <http://knxtoday.com/2014/01/3056/solutions-ip-and-knx-bringing-you-up-to-speed.html>
- [36] VANĚK, Martin. *Řízení provozu budovy pomocí systému KNX*. 2012.

## A Obsah přiloženého CD

- Zdrojové kódy aplikace Hestia (Visual Studio Solution)
- Elektronická verze této práce ve formátu .PDF

## B Uživatelská příručka



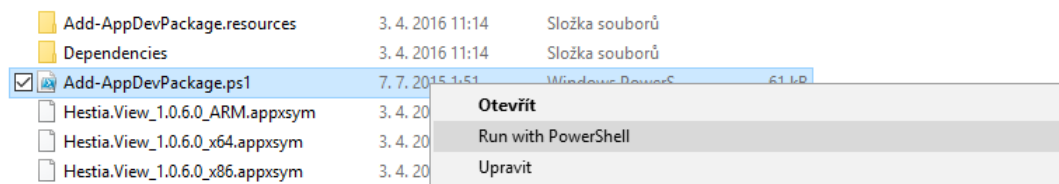
### B.1 Popis aplikace

Na následujících několika stránkách je popsána aplikace Hestia, která slouží k ovládání inteligentní budovy vybavené sběrníkovým systémem KNX. Jedná se o univerzální aplikaci pro operační systém Windows 10 s podporou hlasového ovládání. V uživatelské příručce je popsána instalace aplikace, její konfigurace, možná nastavení a povely pro hlasové ovládání.

V případě ovládání inteligentní instalace z počítače je nutné počítač připojit buď přímo ke KNX/IP-routeru, nebo KNX/IP-router připojit k WiFi routeru a počítač připojit do téže sítě. Ovládání realizované pomocí aplikace v telefonu je možné jen za pomoci připojení k WiFi routeru.

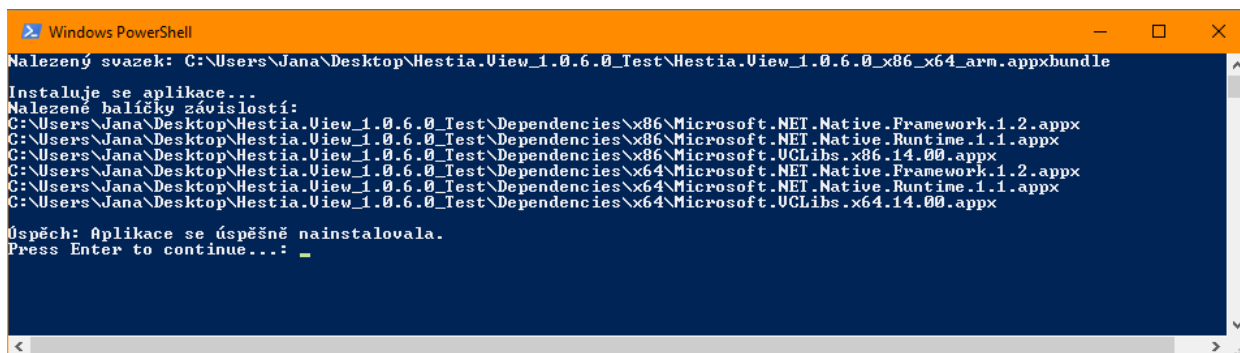
### B.2 Instalace aplikace

Aplikace není k dispozici ve službě Windows Store. Je ji možné nainstalovat z elektronické přílohy. Prvním krokem je přepnutí systému Windows do vývojářského režimu. To lze provést v *Nastavení* -> *Aktualizace a zabezpečení* -> *Pro vývojáře*. Po té je potřeba spustit instalaci aplikace. Ve složce Hestia.View\_1.0.6.0\_Test stačí kliknout pravým tlačítkem na soubor Add-AppDevPackage.ps1 a zvolit "Run with PowerShell".



Obrázek 20: Spuštění instalace

Instalace probíhá automaticky. Po skončení instalace se v konzoli PowerShell zobrazí informace o stavu instalace.



Obrázek 21: Úspěšná instalace aplikace Hestia

Tento způsob instalace je možný v případě instalace na stolní počítače, notebooky nebo tablety. V případě instalování aplikace na Windows 10 mobile je aplikace instalována z počítače do cílového zařízení. Pro úspěšnou instalaci je potřeba učinit několik kroků.

- Před nainstalováním je potřeba zařízení přepnout do vývojářského režimu (postup je popsán výše) a nastavit uložště pro nové aplikace na místní zařízení.
- Připojit mobilní zařízení pomocí USB kabelu k počítači, z něhož bude instalace prováděna.
- Pro instalaci aplikace mimo Windows Store je možné využít WinAppDeployCmd<sup>14</sup>. Jedná se o nástroj příkazové řádky, který je k dispozici po nainstalování Windows 10 SDK.
- V příkazové řádce *cmd.exe* je potřeba otevřít umístění WinAppDeployCmd, to je C:\Program Files (x86)\Windows Kits\10\bin\x86\
- Po té je zadán příkaz pro instalaci aplikace do zařízení. Ten může mít například následující tvar:  
WinAppDeployCmd install -file "C:\Users\Jana\Desktop\Hestia.View\_1.0.6.0\_Test\Hestia.View\_1.0.6.0\_x86\_x64\_arm.appxbundle"-ip 127.0.0.1

### B.3 Hlavní okno aplikace

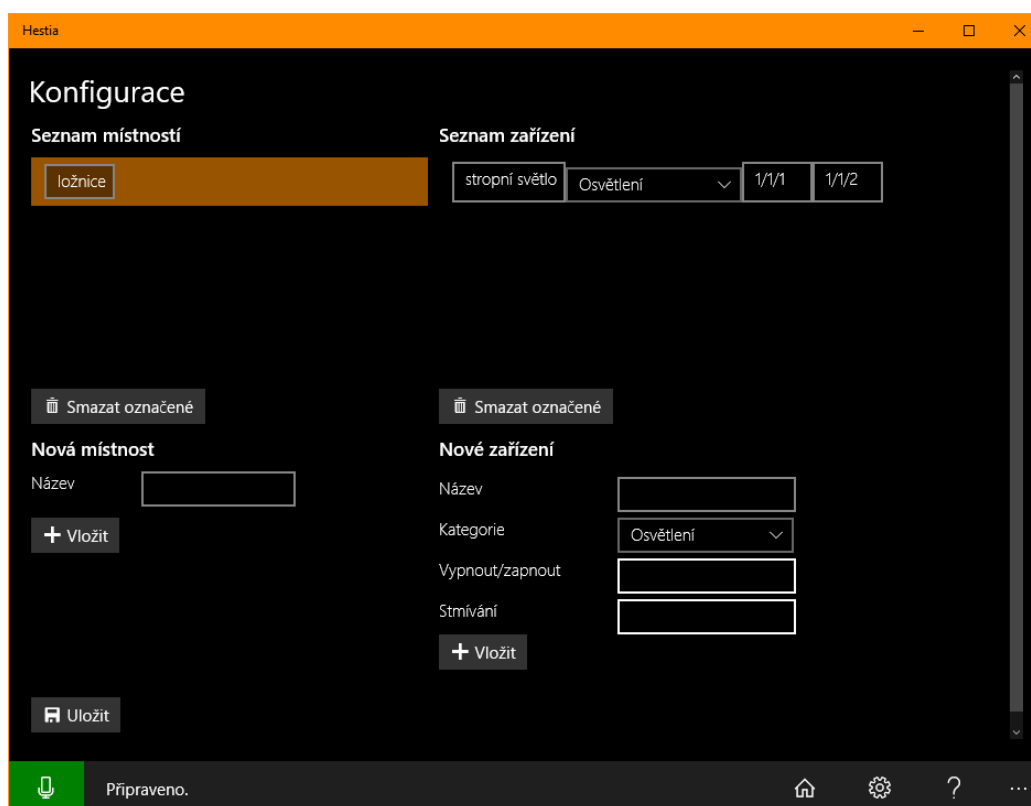
Ve spodní části okna aplikace se nachází navigační lišta s ikonami pro přístup k jednotlivým obrazovkám. Ty se zobrazí v prostoru nad lištou. Panel také obsahuje informaci o stavu rozhraní pro rozpoznání hlasu.

### B.4 Konfigurace

Prvním krokem pro správnou funkčnost aplikace je její konfigurace. Ta se provádí na zvláštní obrazovce, na kterou lze přistoupit z obrazovky nastavení po kliknutí na tlačítko *Otevřít kon-*

<sup>14</sup><https://msdn.microsoft.com/en-us/library/mt203806.aspx?f=255&MSPPErr=-2147217396>

figuraci. V konfiguraci jsou vytvořeny místnosti a zařízení odpovídající skutečnému rozložení v budově. Konfiguraci by měla provádět osoba obeznámena s konfigurací KNX zařízení v budově.



Obrázek 22: Konfigurace aplikace

V levé části obrazovky je zobrazen seznam místností. Místnost lze do seznamu přidat po vyplnění názvu a stisknutí tlačítka vložit pod částí *Nová místnost*. Místnosti slouží pouze k logickému seskupení zařízení. Jakmile jsou vytvořeny požadované místnosti, je možné pro jednotlivé místnosti nakonfigurovat zařízení.

Nejprve se v levé části ze seznamu zařízení vybere konkrétní místnost a pak se v pravé části vyplní informace o zařízení. Je potřeba vyplnit název zařízení a alespoň jednu skupinovou adresu. Skupinové adresy se vyplňují ve tvaru 0-255/0-255/0-255. Konkrétním příkladem může být adresa 1/1/2. Pokud je zadána adresa v nesprávném tvaru, pak je pole s adresou červeně zvýrazněno a adresa nebude uložena.

K dispozici jsou dvě kategorie zařízení a to osvětlení a žaluzie. V případě kategorie osvětlení lze zadat skupinovou adresu pro vypnutí/zapnutí světel a skupinovou adresu pro stmívání. Pokud je vybrána kategorie žaluzie, pak jsou k dispozici pole pro zadání tří adres, a to pro pohyb nahoru/dolů, pro zadání konkrétní výšky žaluzie a naklonění lamel. Zařízení je přidáno do seznamu po stisknutí tlačítka *Vložit*.

Po dokončení celé konfigurace je možné konfiguraci uchovat i pro příští spuštění aplikace. Slouží k tomu tlačítko *Uložit*. Pokud nebude konfigurace uložena, bude k dispozici pouze do

**Nové zařízení**

Název:

Kategorie:

Vypnout/zapnout:

Stmívání:

**+ Vložit**

Obrázek 23: Příklad špatně vyplněné skupinové adresy

zavření aplikace.

## B.5 Ovládání

Samotné ovládání probíhá na zvláštní obrazovce, která se zobrazí po spuštění aplikace jako první. V případě, že neproběhla konfigurace, bude obrazovka prázdná. Jinak se v levé části zobrazí seznam místností a po kliknutí na příslušnou místnost jsou v ní zobrazeny v ní nakonfigurované zařízení. Podle typu zadaných funkcí v konfiguraci se u zařízení zobrazí odpovídající ovládací prvky.



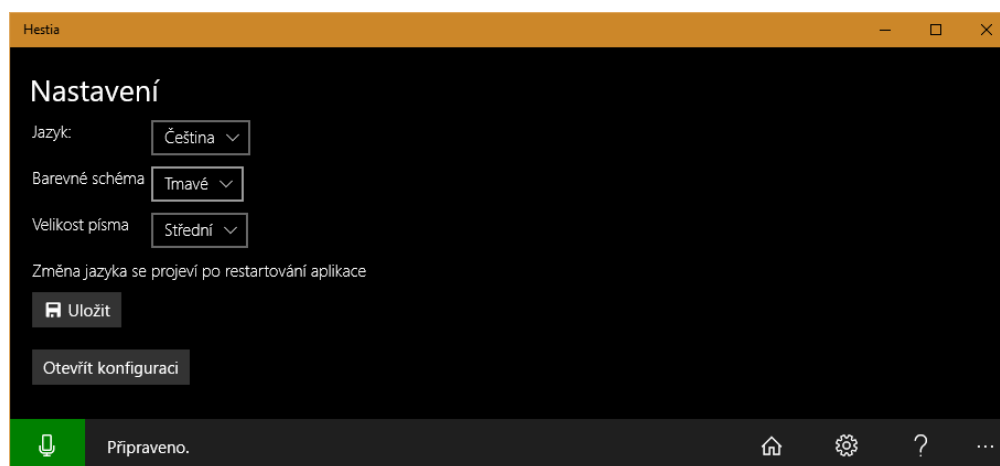
Obrázek 24: Obrazovka ovládání

Pokud se jedná o zařízení typu osvětlení a je v konfiguraci vyplněna skupinová adresa pro zapnutí/vypnutí osvětlení, pak jsou zobrazeny tlačítka *Zapnuto*, *Vypnuto*. Jestliže byla vyplněna adresa pro stmívání, jsou zobrazena tlačítka *+* a *-*.

U typu zařízení žaluzie jsou zobrazovány na základě konfigurace tlačítka *Nahoru* a *Dolů*. Dále vodorovný posuvník pro nastavení náklonu žaluzie. Náklon se nastavuje v rozmezí 0-100%. Pro zadání konkrétní výšky žaluzie je zobrazen svislý posuvník s možností zadání čísla v rozmezí 0-100%. Pro nastavení náklonu lamel žaluzie nebo výšky žaluzie je nutné potvrzení tlačítkem *OK*.

## B.6 Nastavení

V části s nastavením je možné změnit několik parametrů aplikace. Na výběr je mezi českým a anglickým jazykem. Dále je možné změnit barevné schéma, k dispozici je světlé a tmavé. Pro zlepšení čitelnosti textů v aplikaci lze vybrat mezi třemi velikostmi písma.



Obrázek 25: Náhled obrazovky s nastavením

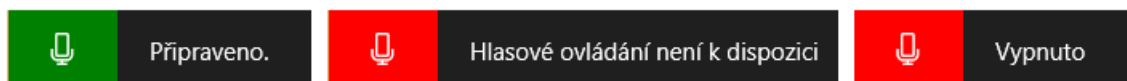
V případě změny písma nebo barevného schéma je změna patrná ihned, ale pro uchování zvoleného nastavení i pro příští spuštění aplikace, je potřeba nastavení uložit tlačítkem *Uložit*. Změna jazyka aplikace se projeví až po opětovném spuštění.

K dispozici je také tlačítko *Otevřít konfiguraci*, které slouží k otevření obrazovky s konfigurací.

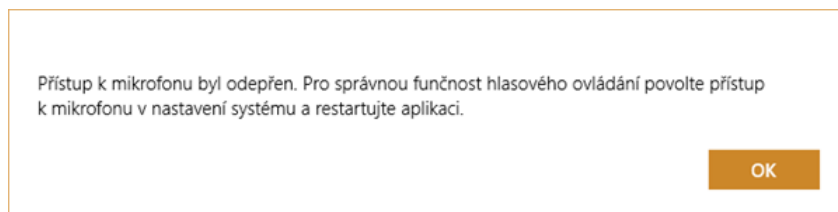
## B.7 Hlasové ovládání

Aplikace podporuje hlasové ovládání v anglickém jazyce. Je nutné aplikaci umožnit přístup k mikrofonu, v opačném případě není možné hlasové ovládání využít. Pokud není umožněn přístup k mikrofonu, pak se po spuštění zobrazí informační okno. V navigační liště se zobrazí červená ikona mikrofonu. Aplikace je plně funkční i bez hlasového rozhraní. Hlasové rozhraní je možné vypnout po kliknutí na ikonu mikrofonu. Barva ikony se změní na červenou.

Na každé obrazovce jsou k dispozici různé povely. Navigační povely jsou k dispozici v celé aplikaci. Vždy je nejprve vyřčen povel *Go to* a po té název obrazovky. Za názvem obrazovky je možné ještě uvést slovo *page*, to je však nepovinné.



Obrázek 26: Stavby hlasového rozhraní.



Obrázek 27: Upozornění při nepovolení přístupu k mikrofonu

Tabulka 3: Navigační povely

hlasový povel	akce
go to control page	navigace na ovládací obrazovku
go to settings page	navigace na obrazovku s nastavením
go to help page	navigace na obrazovku s nápovědou

Při přepnutí na obrazovku s ovládáním lze využít povely pro výběr místnosti v levém panelu pomocí příkazu *Select room*, za kterým je uvedeno číslo v rozmezí 1-20. Stejným způsobem lze vybrat zařízení pomocí příkazu *Select device*. Po výběru zařízení je možné v případě osvětlení ovládat zapínání a vypínání světel. Slouží k tomu příkazy *light on* a *light off*. Ovládat stmívací člen a zvyšovat jeho jas pomocí příkazů *increase brightness* a *decrease brightness*.

Tabulka 4: Příkazy pro ovládání budovy

hlasový povel	akce
select room <i>five</i>	výběr páté místnosti v pořadí
select device <i>twelve</i>	výběr dvanáctého zařízení v pořadí
light on	zapnutí světla
light off	vypnutí světla
blind up	pohyb žaluzie nahoru
blind down	pohyb žaluzie dolů
write height value <i>seven five</i>	nastavení hodnoty posuvníku pro výšku žaluzie na 75
change blind height	potvrzení změny výšky žaluzie
write tilt value <i>seven five</i>	nastavení hodnoty posuvníku pro náklon lamel žaluzie na 75
change tilt	potvrzení změny náklonu lamel

Pokud se jedná o zařízení typu žaluzie, pak je k dispozici hlasové ovládání pohybu nahoru a dolů pomocí příkazů *blind up* a *blind down*. Pro nastavení výšky žaluzie lze použít příkaz *write height value* následováno zadáním čísla v rozmezí 0-100 pomocí jednotlivých číslovek. Pro zadání čísla 50 bude vyřčeno *five zero*. Po zadání hodnoty posuvu je akce posuvu vyvolána příkazem



*change blind height*. Pro nastavení hodnoty naklonění lamel slouží příkaz *write tilt value* a po něm číslo stejně jako v předchozím případě. Vykonání změny náklonu lamel je potvrzeno příkazem *change tilt*

Hlasové ovládání lze využít také na obrazovce s nastavením aplikace, kde je možné příkazem *change language to* změnit jazyk na Češtinu *Czech*, nebo Angličtinu *English*. Pro nastavení velikosti písma aplikace se používá příkaz *change font size to* následovaný jednou z velikostí *small* (malé), *medium* (střední), *large* (velké). Poslední nastavení, které lze změnit, je barevné schéma aplikace. To se provádí pomocí příkazu *change theme to*. Nastavení lze uložit příkazem *save settings* a pomocí povelu *open configuration* lze otevřít konfiguraci aplikace.

Tabulka 5: Hlasové příkazy pro nastavení aplikace

hlasový povel	akce
<i>change language to Czech</i>	výběr českého jazyka aplikace
<i>change font size to medium</i>	změna velikosti písma na střední
<i>change theme to dark</i>	změna barevného schéma aplikace na tmavé
<i>save settings</i>	uložení nastavení aplikace
<i>open configuration</i>	otevření obrazovky s konfigurací

Poslední obrazovkou, kterou lze ovládat hlasovými povely, je obrazovka s konfigurací aplikace. Je možné hlasem vytvořit novou místnost. Jméno místnosti je vloženo příkazem *write room name* následované jedním z možných názvů ze seznamu níže, název místnosti lze doplnit volitelně číslem v rozmezí 1-9. Místnost je vložena příkazem *add new room*.

- kitchen - kuchyně
- bathroom - koupelna
- toilet - toaleta
- living room - obývací pokoj
- dinning room - jídelna
- bedroom - ložnice
- corridor - chodba
- hall - hala
- basement - sklep
- garage - garáž
- attic - podkroví

Pro vložení názvu nového zařízení se využívá příkaz *write device name* a některý z názvu v seznamu níže, který lze také doplnit číslem v rozmezí 1-9.

- blind - žaluzie
- north blind - severní žaluzie
- south blind - jižní žaluzie
- east blind - východní žaluzie
- west blind- západní žaluzie
- rear blind - zadní žaluzie
- front blind - přední žaluzie
- left blind - levá žaluzie
- right blind - pravé žaluzie
- ceiling light - stropní světlo
- light - světlo
- lamp - lampa

Pro změnu kategorie nového zařízení se využívá příkaz *write device category* a jedna z možností *blinds*(žaluzie), nebo *lights*(osvětlení). Pro zápis skupinové adresy slouží příkaz *write lights on off address*, kde se na základě konkrétního typu funkce adresy udává některé z možností *lights on off* (vypínání a zapínání světel), *lights dimming* (stmívání), *blinds up down* (pohyb žaluzií nahoru a dolů), *blinds value*(posuv do konkrétní pozice), *blinds slat* (naklonění lamel žaluzie). Po té je zadáno číslo nadiktováním jednotlivých číslic oddělených lomítkem. Nové zařízení je pak do seznamu vloženo příkazem *add new device*.

Pro změnu hodnot některé již vytvořené místnosti nebo zařízení stačí zaměnit v příkazech slovo *write* za *change*. Je však nutné vybrat danou místnost nebo zařízení pomocí příkazu *select room* nebo *select device*, který byl popsán dříve. Celou konfiguraci lze uložit příkazem *save configuration*.

Tabulka 6: Hlasové příkazy pro konfiguraci aplikace

hlasový povel	akce
write room name <i>kitchen</i>	vložení názvu místnosti s textem kitchen
change room name <i>kitchen</i>	změna textu názvu místnosti na kitchen
write device name <i>blind nine</i>	vložení názvu zařízení s textem blind 9
change device name <i>blind nine</i>	změna textu názvu zařízení na blind 9
write light dimming address <i>one slash seven five slash nine</i>	vložení adresy 1/75/9 pro stmívání
change light dimming address <i>one slash seven five slash nine</i>	změna adresy pro stmívání na 1/75/9
write device category <i>lights</i>	výběr osvětlení pro kategorii nového zařízení
change device category <i>lights</i>	změna kategorie nového zařízení na osvětlení
add new room	vložení nové místnosti
add new device	vložení nové místnosti
save configuration	uložení konfigurace
select room <i>seven</i>	výběr sedmé místnosti ze seznamu
select device <i>four</i>	výběr čtvrtého zařízení ze seznamu